



Introduction To DevOps

Lesson 01 – What is DevOps

Lesson Objectives

- DevOps – the business need
- What is DevOps?
- How does DevOps Work?
- DevOps Practices
 - Source Code Management
 - Automated testing
 - Containerization
 - Continuous integration
 - Configuration Management
 - Automated Monitoring
 - Continuous Deployment and Release Management



DevOps – the business need



The Developer



Time to Market



Dependency Error

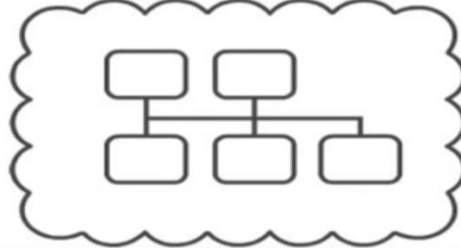
DevOps – the business need



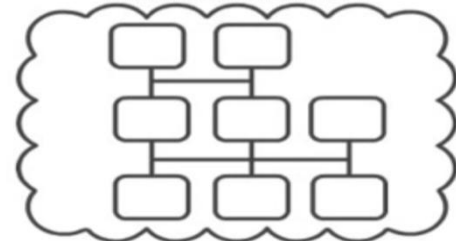
The Developer

As a developer I have always dabbled lightly in operations. I always wanted to focus on making my code great and let an operations team worry about setting up the production infrastructure.

**DEVELOPMENT
ENVIRONMENT**



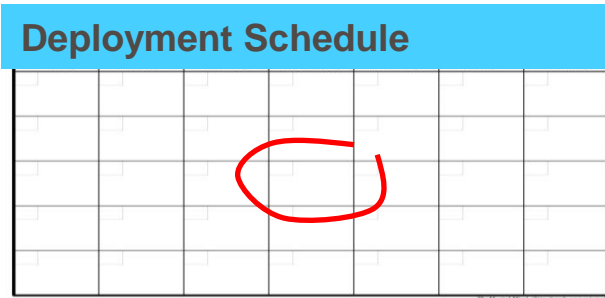
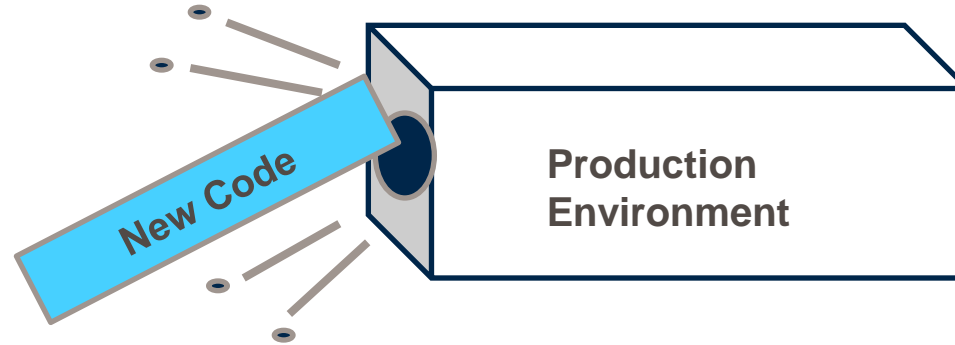
**PRODUCTION
ENVIRONMENT**



DevOps – the business need



The Operations team



DevOps – the business need



The Operations team

I am responsible for maintaining 99% uptime. I think of servers and new code deployment mostly introduces bugs which I need to fix to ensure availability. These developers are pushing their work to me.



New Code



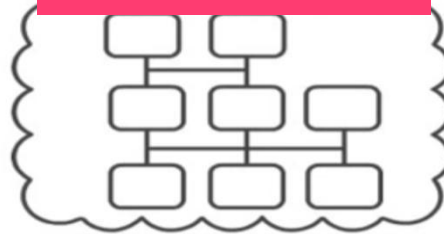
DevOps – the business need

DevOps

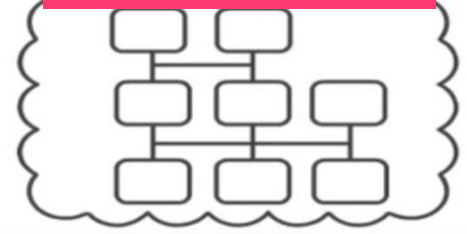


- ✓ **Worked Better together**
- ✓ **Thought more alike**
- ✓ **Broke down silos**
- ✓ **Shared responsibilities?**

**DEVELOPMENT
ENVIROMNENT**



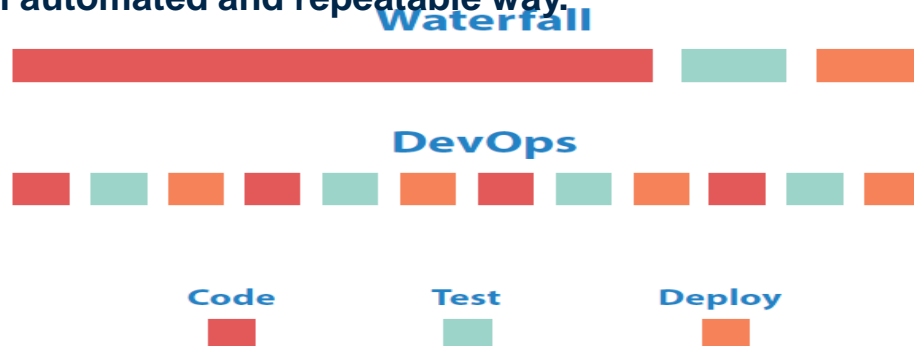
**PRODUCTION
ENVIROMNENT**



What is DevOps?

The Definition:

- ✓ “a software development method that stresses communication, collaboration & integration between software developers and IT professionals.” - wikipedia
- ✓ “DevOps is simply operations working together with engineers to get things done faster in an automated and repeatable way.”



C.A.L.M.S.

C – Culture

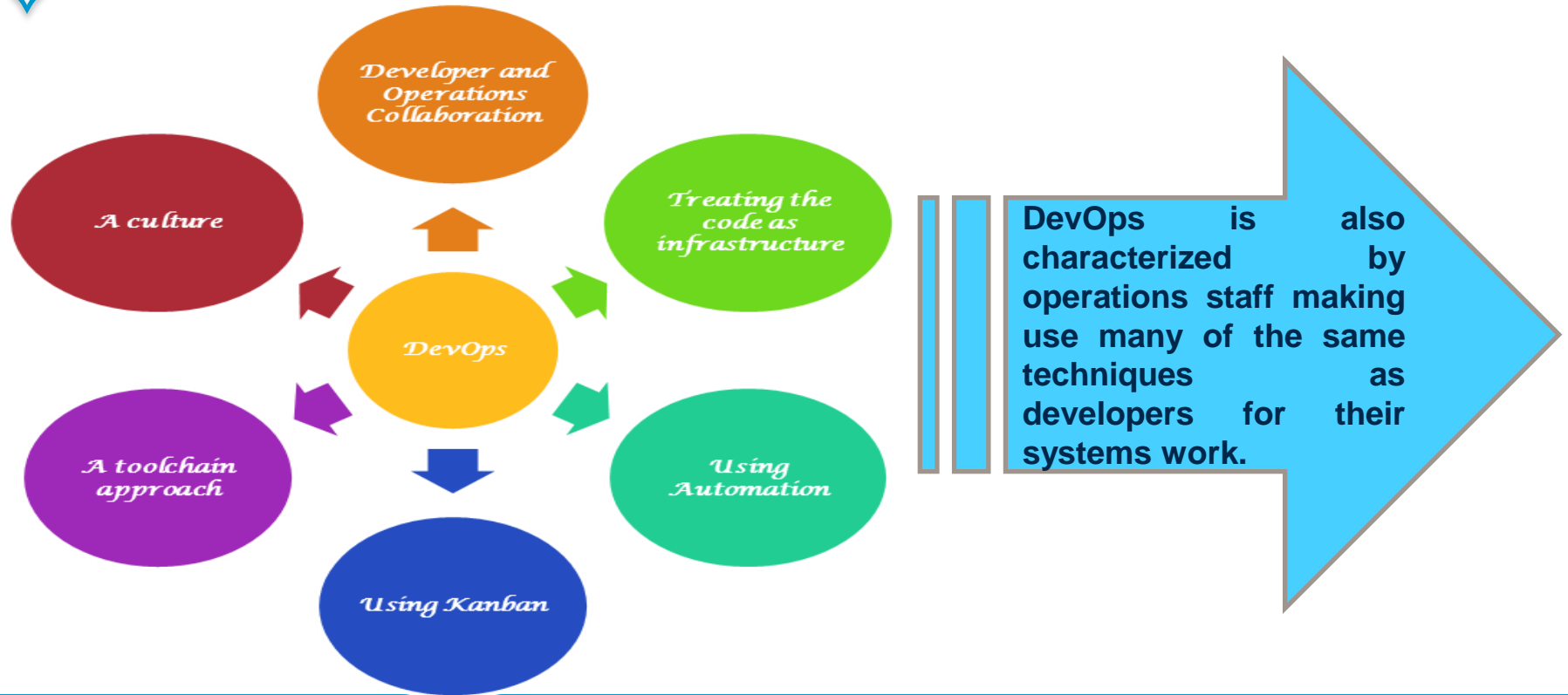
A – Automation

L – Lean

M – Measurement

S – Sharing

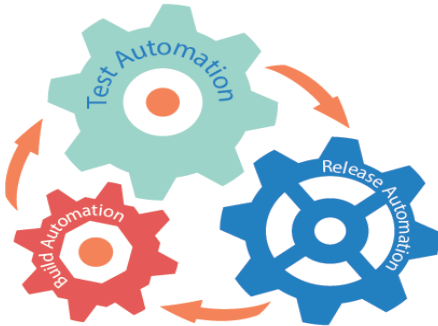
What is DevOps?



What is DevOps?

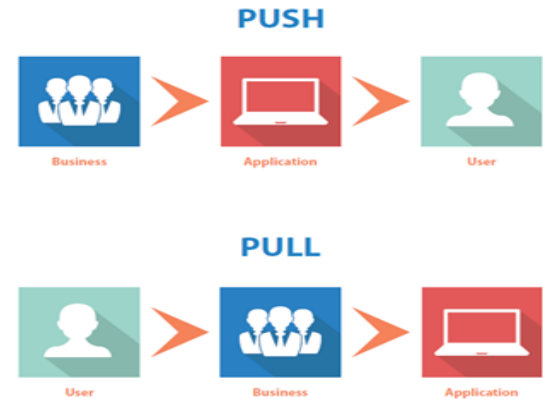
Automation - Optimizing the Entire Pipeline

- ✓ The best way to quicken processes across the pipeline is to automate them.
- ✓ Build automation can be approached using Continuous Integration (CI) tools like Jenkins.
- ✓ Test automation requires frameworks like Selenium and Appium.
- ✓ And release automation, which is still maturing, can be handled with tools like Automic.
- ✓ DevOps is about optimizing processes across the entire pipeline



PUSH vs PULL

- ✓ The Lean approach to building apps involves a pull system where customers define what you should focus on, how fast you should go, and what you should ship, as opposed to the traditional top-down model of building applications.



What DevOps is Not?

- **It's Not NoOps :**

DevOps is not that Developers take over Ops!

- **It's Not (Just) Tools:**

DevOps is also not simply implementing a set of tools.

- **It's Not (Just) Culture**

DevOps consists of items at all the levels

- **It's Not (Just) Devs and Ops**

What about security people! And network admins!

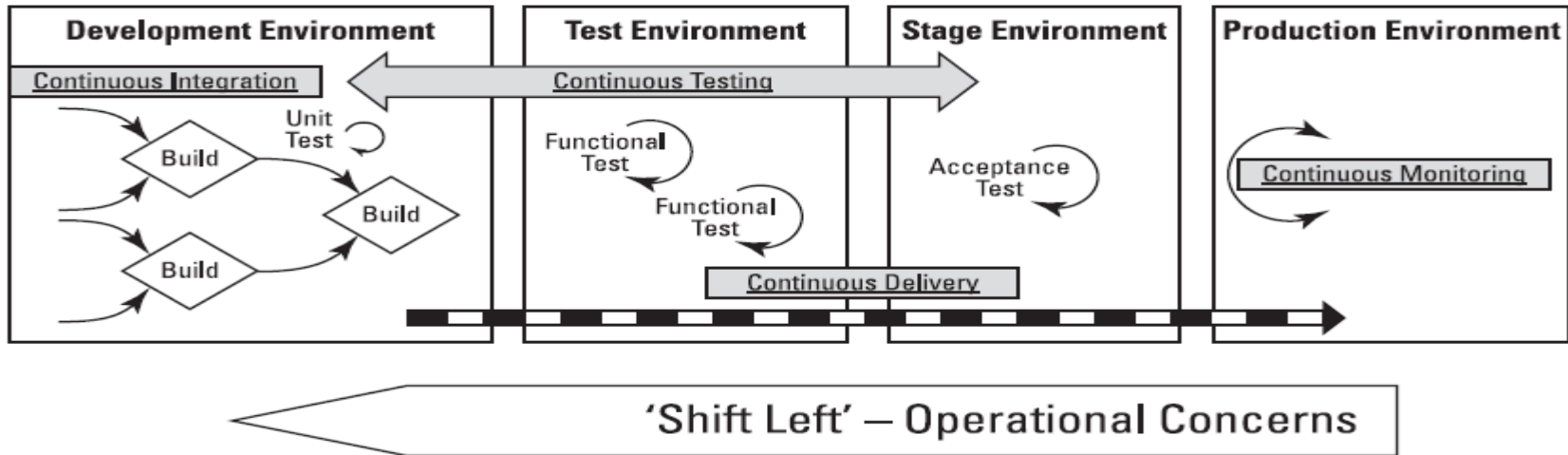
- **It's Not Everything**

It is part of an overall, hopefully collaborative and agile corporate culture, but DevOps is specifically about how operations plugs into that

How does DevOps Work?

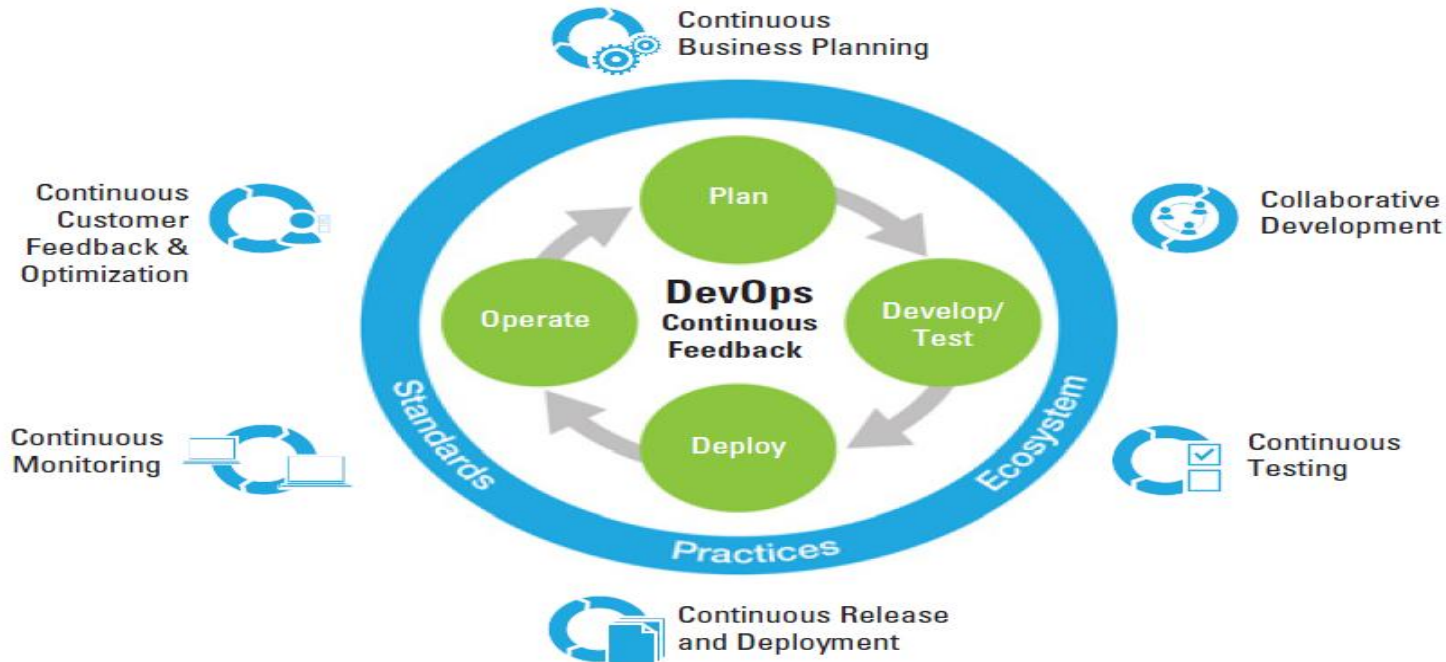
- **The 4 principles:**

- ✓ Develop and test against production-like systems
- ✓ Deploy with repeatable, reliable processes
- ✓ Monitor and validate operational quality
- ✓ Amplify feedback loops



How does DevOps Work?

The Reference Architecture:



How does DevOps Work?

The Reference Architecture:

- **Plan:**

Focuses on establishing business goals and adjusting them based on customer feedback: continuous business planning .

- **Develop/Test:**

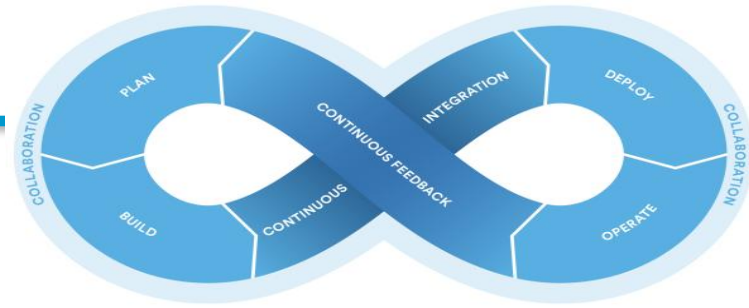
Forms the core of development and quality assurance (QA) capabilities. It involves two practices - collaborative development and continuous testing.

- **Deploy**

Continuous release and deployment take the concept of continuous integration to the next step

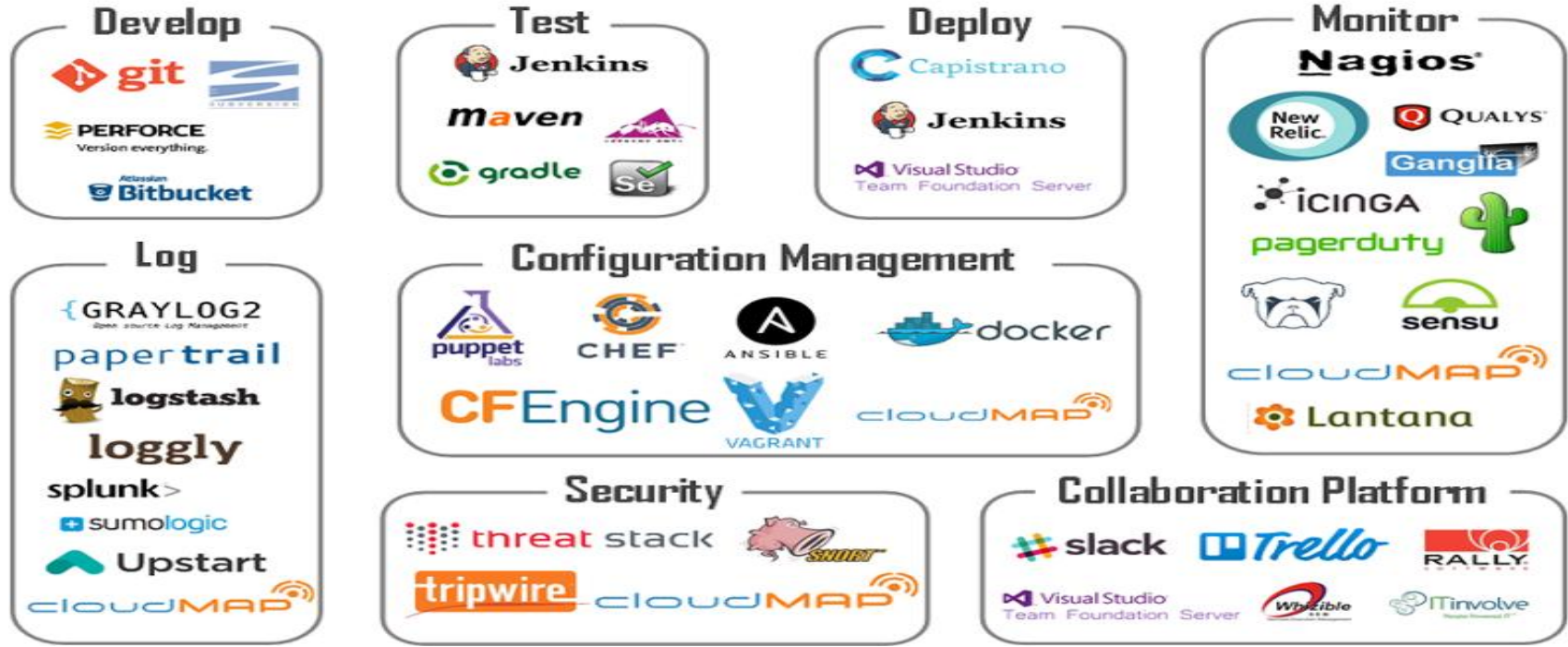
- **Operate**

It involves two practices - continuous monitoring and continuous customer



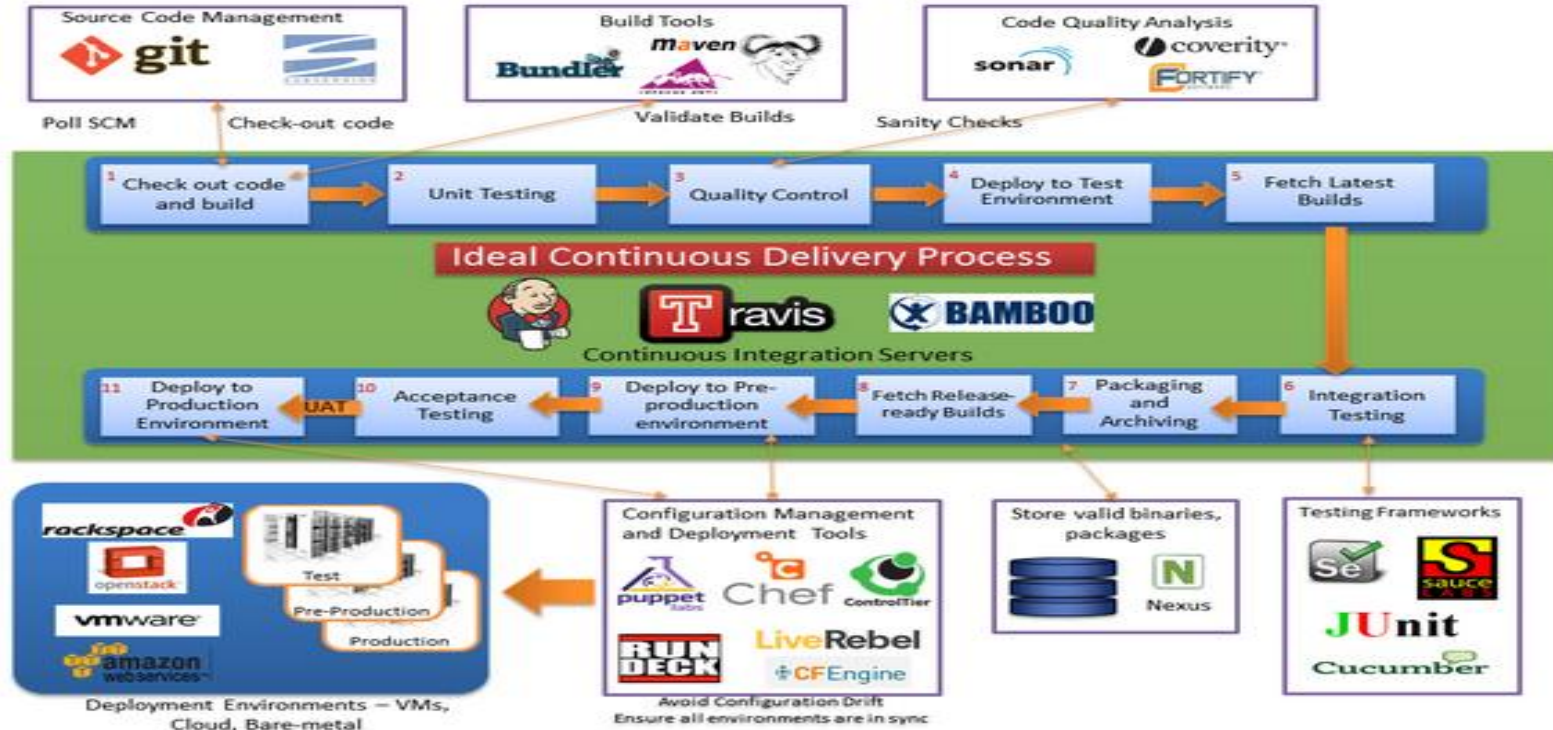
How does DevOps Work?

No tool will magically make the team DevOps.



How does DevOps Work?

The Continuous Delivery Pipeline



DevOps Practices

- Source Code Management
- Automated testing
- Containerization
- Continuous integration
- Configuration Management
- Automated Monitoring

Source Code Management

- Continually merges source code updates from all developers on a team into a shared mainline.
- A source code manager (SCM) is a software tool used by teams of programmers to manage source code.
- SCMs are used to track revisions in software.
- Each revision is given a timestamp and includes the name of the person who is responsible for the change.
- Various revisions may be compared, stored, and merged with other revisions.

Example : GIT

GIT

<https://git-scm.com/>



The screenshot shows the Git website homepage. At the top left is the Git logo (a red diamond with a white 'T') followed by the text 'git --distributed-even-if-your-workflow-isnt'. To the right is a search bar with the placeholder text 'Search entire site...'. Below the header, there are two paragraphs of text. The first paragraph describes Git as a free and open source distributed version control system. The second paragraph describes Git as easy to learn and having a tiny footprint with lightning fast performance. Below the text is a section titled 'Learn Git in your browser for free with Try Git.' with a small Git logo. To the right of the text is a diagram showing a network of stacks of papers connected by colored lines (red, green, yellow). Below the text and diagram is a section with four icons and text: 'About' (gear icon), 'Documentation' (book icon), 'Downloads' (download icon), and 'Community' (speech bubble icon). To the right of this section is a monitor displaying the 'Latest source Release 2.9.3' and 'Release Notes (2016-08-12)' with a button for 'Downloads for Windows'.

git --distributed-even-if-your-workflow-isnt

Git is a **free and open source** distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is **easy to learn** and has a **tiny footprint with lightning fast performance**. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like **cheap local branching**, convenient **staging areas**, and **multiple workflows**.

Learn Git in your browser for free with Try Git.

About
The advantages of Git compared to other source control systems.

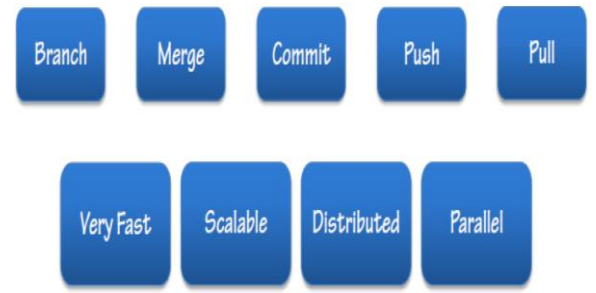
Documentation
Command reference pages, Pro Git book content, videos and other material.

Downloads
GUI clients and binary releases for all major platforms.

Community
Get involved! Bug reporting, mailing list, chat, development and more.

Latest source Release
2.9.3
Release Notes (2016-08-12)
Downloads for Windows

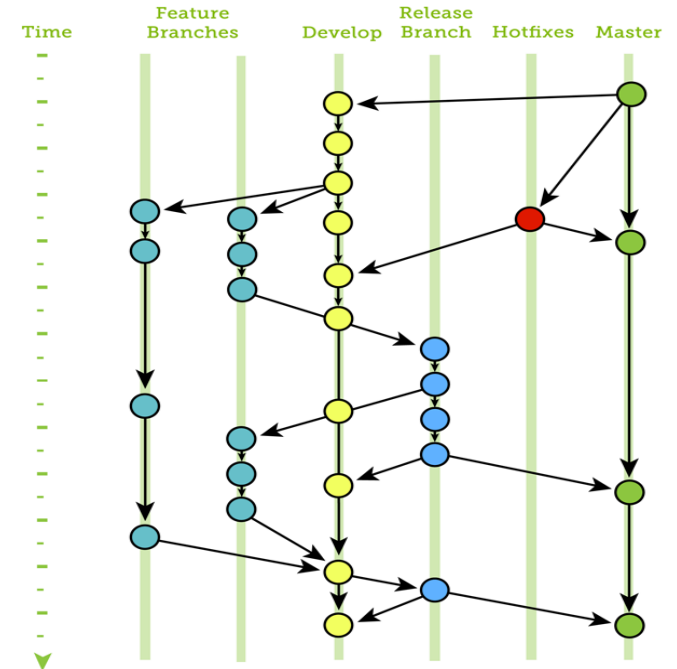
Distributed Version
Control System
("DVCS")



GIT

Distributed Version Control using GIT

- Git is a distributed version control system.
- A distributed version control system does not necessarily have a central server which stores the data.
- The user can copy an existing repository. This copying process is typically called cloning
- Git allows the user to synchronize the local repository with other (remote) repositories.
- Users with sufficient authorization can push changes from their local repository to remote repositories.
- They can also fetch or pull changes from other repositories to their local Git repository.

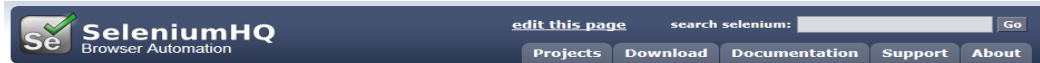


Automated Testing

- The objective of automated testing is to simplify as much of the testing effort as possible with a minimum set of scripts.
- Automated testing tools are capable of executing repeatable tests, reporting outcomes, and comparing results with faster feedback to the team.
- Automated tests perform precisely the same operation each time they are executed, thereby eliminating human errors – and can be run repeatedly, at any time of day.
- Includes testing for each environment in the pipeline
 - Dev. Environment
 - Unit, Sanity Testing
 - CI Environment
 - Incremental Integration Testing
 - QA Environment
 - Functional , Usability Testing
 - Compatibility Testing

Selenium

<http://docs.seleniumhq.org/>



What is Selenium?

Selenium automates browsers. That's it! What you do with that power is entirely up to you. Primarily, it is for automating web applications for testing purposes, but is certainly not limited to just that. Boring web-based administration tasks can (and should!) also be automated as well.

Selenium has the support of some of the largest browser vendors who have taken (or are taking) steps to make Selenium a native part of their browser. It is also the core technology in countless other browser automation tools, APIs and frameworks.

Which part of Selenium is appropriate for me?

Selenium WebDriver



If you want to

- create robust, browser-based regression automation suites and tests
- scale and distribute scripts across many environments

Selenium IDE



If you want to

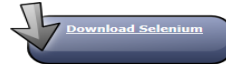
- create quick bug reproduction scripts
- create scripts to aid in automation-aided exploratory testing



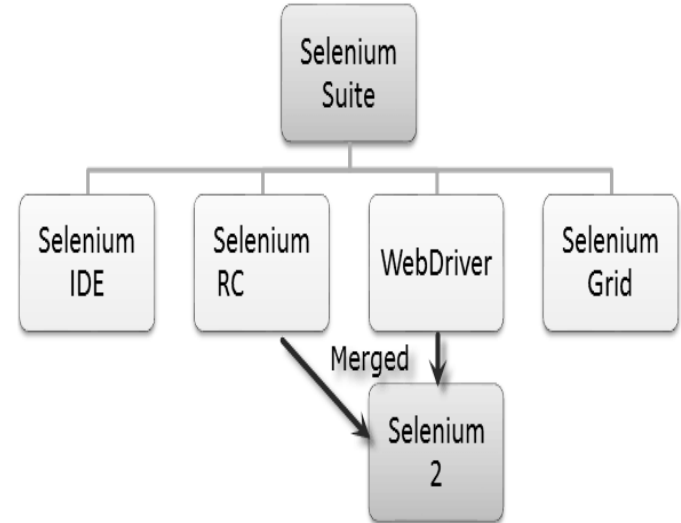
Selenium is a suite of tools to automate web browsers across many platforms.

Selenium...

- runs in [many browsers](#) and [operating systems](#)
- can be controlled by many [programming languages](#) and [testing frameworks](#).



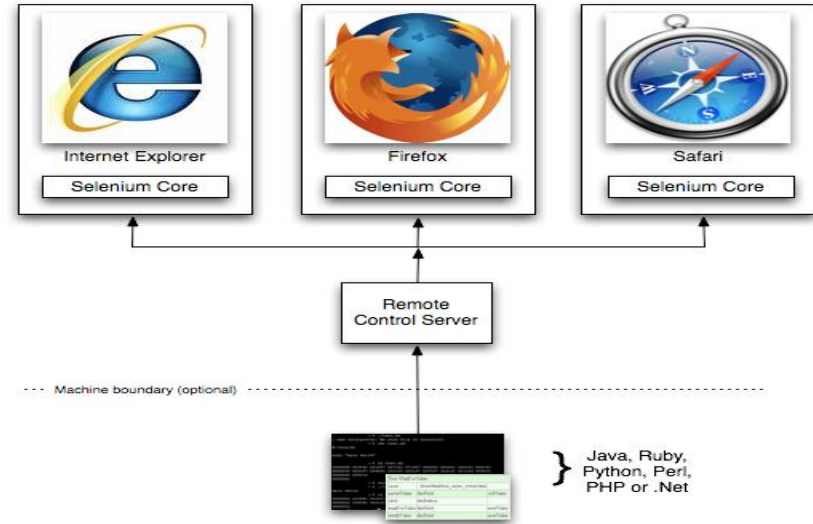
[Donate to Selenium](#)



Selenium

Overview of Selenium IDE

- Allows you to record, play back, edit, and debug tests in browser.
- Generate scripts from recorded user actions in most of the popular languages like Java, C#, Perl, Ruby etc.
- Run them using Selenium Web Driver.
- Allows the user to pick from a list of assertions and verifications for the selected location
- Selenium Remote Control (RC) is a test tool that allows you to write automated web application UI tests in any programming language against any HTTP website using any mainstream JavaScript-enabled browser.



Containerization

■ Before:

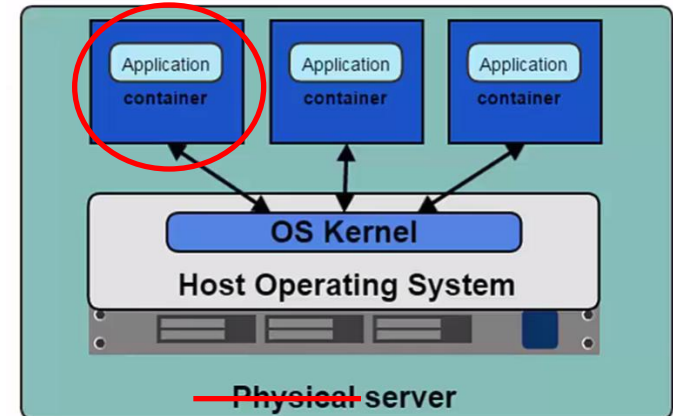
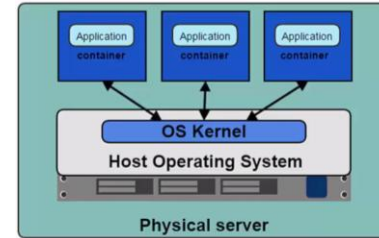
- monolithic applications
- long development cycles
- slowly scaling up

■ Now:

- decoupled services
- fast, iterative improvements
- quickly scaling out

■ Deployment becomes very complex

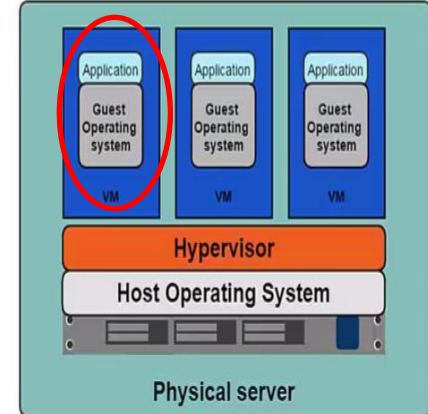
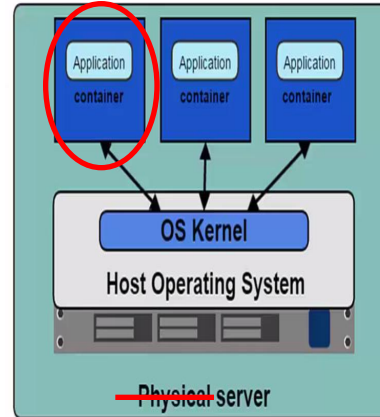
- Many different stacks.
- Many different targets.



Containerization

Containers vs VMs

- Containers are more lightweight
- No need to install guest OS
- Less CPU, RAM, storage space required
- More containers per machine than VMs
- Greater portability

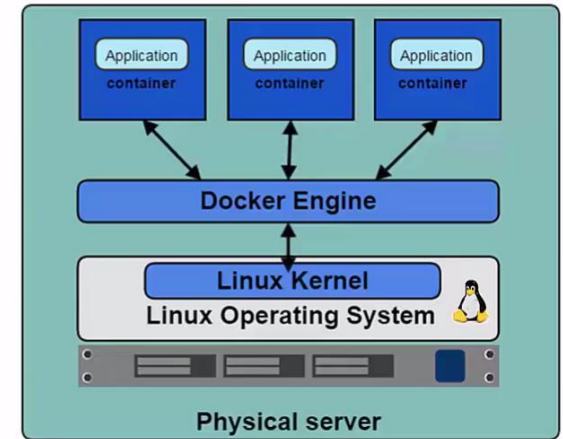


Docker

- The initial container engine is now known as "Docker Engine."
 - Other tools have been added:
 - Docker Compose (formerly "Fig")
 - Docker Machine
 - Docker Swarm
 - Kitematic (acquisition)
 - Tutum (recent acquisition)

Docker and the Linux Kernel

- **Docker Engine** is the program that enables containers to be built, shipped and run.
- Docker Engine uses Linux Kernel namespaces and control groups
- Namespaces give us the isolated workspace



Continuous Integration (CI)



Fail Fast

- Continually merges source code updates from all developers on a team into a shared mainline.
- Prevents a developer's local copy of a software project from drifting too far afield as new code is added by others, avoiding catastrophic merge conflicts.
- CI involves a centralized server that continually pulls in all new source code changes as developers commit them and builds the software application from scratch, notifying the team of any failures in the process.
- If a failure is seen, the development team is expected to refocus and fix the build before making any additional code changes.

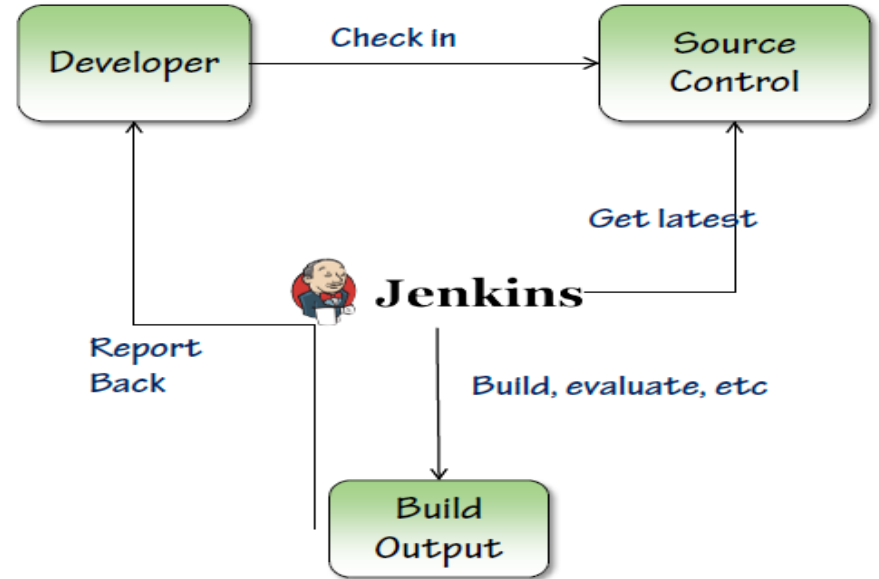
Example : Jenkins / Bamboo / Go

Jenkins

<https://jenkins.io>



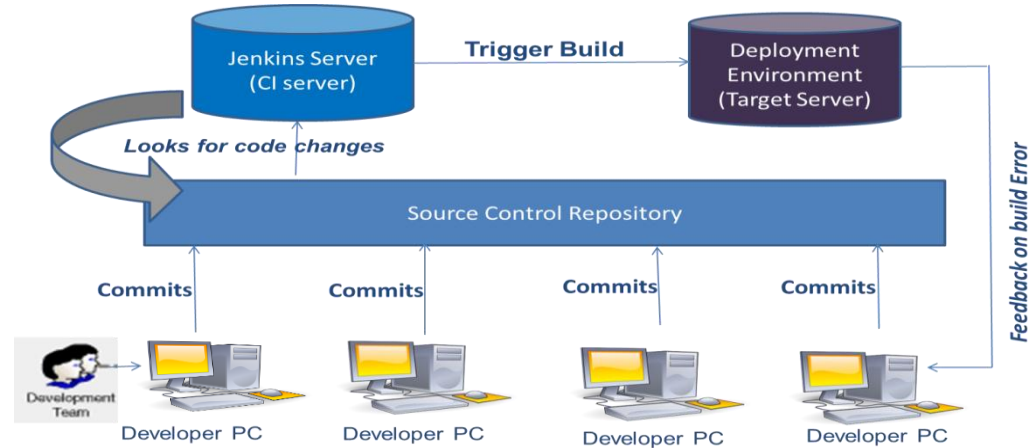
The screenshot shows the Jenkins website homepage. At the top is a navigation bar with links: Plugins, Use-cases, Participate, Sub-projects, Resources, Security, Press, and Conduct. Below the navigation bar is a large blue banner featuring the Jenkins logo (a cartoon man in a tuxedo) on the left. To the right of the logo, the word "Jenkins" is written in large white letters, followed by the tagline "Build great things at any scale". Below this, a paragraph states: "The leading open source automation server, Jenkins provides hundreds of plugins to support building, deploying and automating any project." A red button labeled "Download Jenkins" is positioned below the text. At the bottom of the banner, it says "Get 2.7.2 LTS .war or the latest 2.18 weekly release".



Jenkins

Open Source CI Tool:

- Jenkins is an open source continuous integration tool written in java developed by Kohsuke Kawaguchi.
- Monitors the change in the source control systems like SVN, CVS, etc.
- Builds the application using various build tools like ANT, MAVEN, etc.
- Provides a fresh build whenever there is a change in the source control system
- Sends messages on the status of the build through Email, SMS, etc.



- Can support software releases, documentation, monitoring, and a number of use case secondary to continuous integration

Configuration Management Tool – Chef and Puppet

<https://www.chef.io/chef/>



CHEF SERVER

The Chef server is the highly scalable foundation of the Chef automation platform. Create a flexible, dynamic infrastructure across multiple datacenters, public and private clouds, and in heterogeneous environments.

[Download >](#) [Learn More >](#)



Automate

Write dynamic policies that automatically create and configure infrastructure when you need it



Search

Search your entire infrastructure at any time—and use real-time data in your policies



Deliver

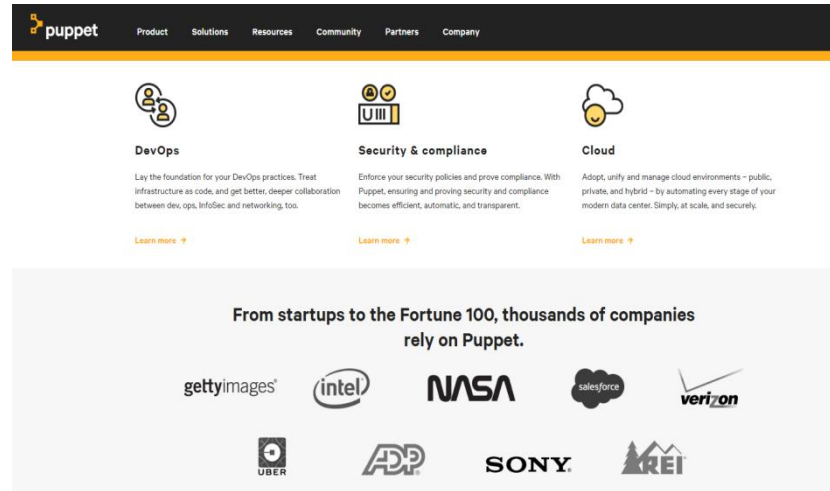
Automatically deliver the latest tested and approved policies to your infrastructure



Scale

Manage complexity with the most scalable automation platform on the planet

<https://puppet.com>



puppet Product Solutions Resources Community Partners Company

DevOps
Lay the foundation for your DevOps practices. Treat infrastructure as code, and get better, deeper collaboration between dev, ops, InfoSec and networking, too.
[Learn more >](#)

Security & compliance
Enforce your security policies and prove compliance. With Puppet, ensuring and proving security and compliance becomes efficient, automatic, and transparent.
[Learn more >](#)

Cloud
Adopt, unify and manage cloud environments - public, private, and hybrid - by automating every stage of your modern data center. Simply, at scale, and securely.
[Learn more >](#)

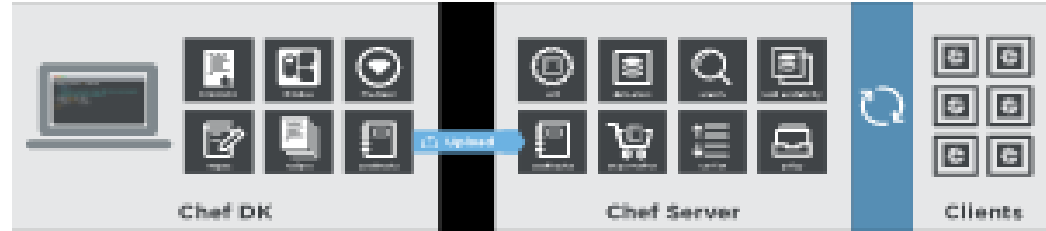
From startups to the Fortune 100, thousands of companies rely on Puppet.

gettyimages® intel NASA salesforce verizon

UBER ADP SONY KREI

Chef

- Express your infrastructure policy – how your software is delivered and maintained on your servers – as code.
- The normal Chef workflow involves managing servers remotely from your workstation.
- A Chef resource describes some piece of infrastructure, such as a file, a template, or a package.
- A Chef recipe is a file that groups related resources, such as everything needed to configure a web server, database server, or a load balancer.



1. Install IIS

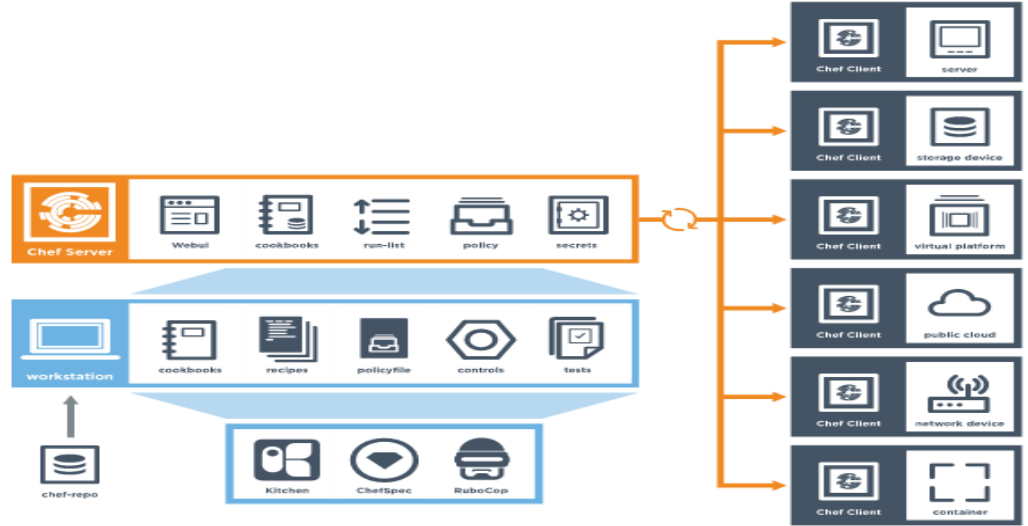
Let's install IIS. From your `~\chef-repo` directory, add this recipe to a file named `webserver.rb`.

```
Editor: ~\chef-repo\webserver.rb
1 powershell_script 'Install IIS' do
2   code 'Add-WindowsFeature Web-Server'
3   guard_interpreter :powershell_script
4   not_if "(Get-WindowsFeature -Name Web-Server).Installed"
5 end
```

Chef

Relationships between the various elements of Chef:

- Includes the nodes, the server, and the workstation.
- These elements work together to provide the chef-client the information and instruction that it needs so that it can do its job.



workstation



node



Chef Client



Chef Server



Chef Supermarket



cookbook

Puppet

- Lets you define the desired state of your infrastructure and what you want it to do.
- Puppet automatically enforces that desired state and remediates any unexpected changes.
- Deploy faster, with greater reliability, because one no longer have to map out and manually deploy every step
- **Capabilities:**
 - Orchestration
 - Automated provisioning
 - Configuration automation
 - Visualization & reporting
 - Code management
 - Node management
 - Role-based access control

Deliver faster with a proven DevOps platform

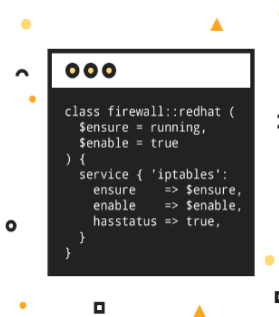
Automation — the foundation for many DevOps practices — helps you move faster without sacrificing stability or security. Now is the time to take advantage of automation and proven DevOps practices to drive your team — and your deployments — forward.

Puppet Enterprise lets you deliver technology changes faster, release better software, and do it all more frequently with confidence.

[Download the DevOps Resource Kit](#)

Lay the foundation for DevOps practices

Puppet Enterprise manages **infrastructure as code**, providing the foundation for DevOps practices such as versioning, **automated testing** and **continuous delivery**. You deploy changes with confidence and recover more quickly from failures, freeing your team to be more agile and responsive to business needs.



Automated Monitoring

- In DevOps, automation monitoring often takes the spotlight
- There are many products that promote monitoring.
- Monitoring predates DevOps, and has evolved as well.
- In DevOps we can write infrastructure as code, automating integration and testing, and deploying everything in the cloud.
- The pace of development has increased,so has the customer feedback loop and deployment tooling.
- As a result there is more to monitor, so where we use DevOps-style tooling to automate integration, testing, provisioning, and deployment, we need to use DevOps-style tooling to monitor our builds, resources, and performance.

Automated Monitoring tools

Automation monitoring includes

- Monitoring development milestones
- Monitoring vulnerabilities
- Deployment monitoring
- Application log monitoring
- Server health monitoring
- Activity Monitoring

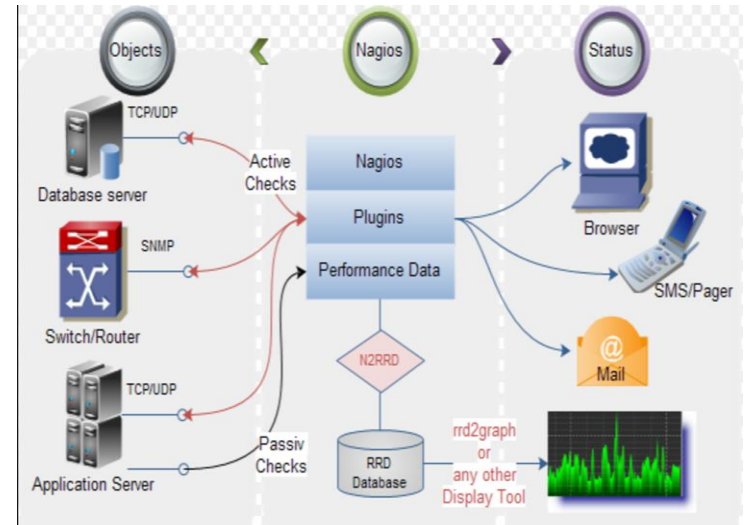
Monitoring tools

	Nagios
	AWS Cloudwatch
	New Relic
	Home-grown
	Icinga
	Sensu
	Zabbix
	DataDog
	Riemann
	SCOM
	Zenoss
	Centreon
	Librato
	Xymon
	Opsview

Nagios

Nagios Core is open source software licensed under the GNU GPL V2 and it provides ...

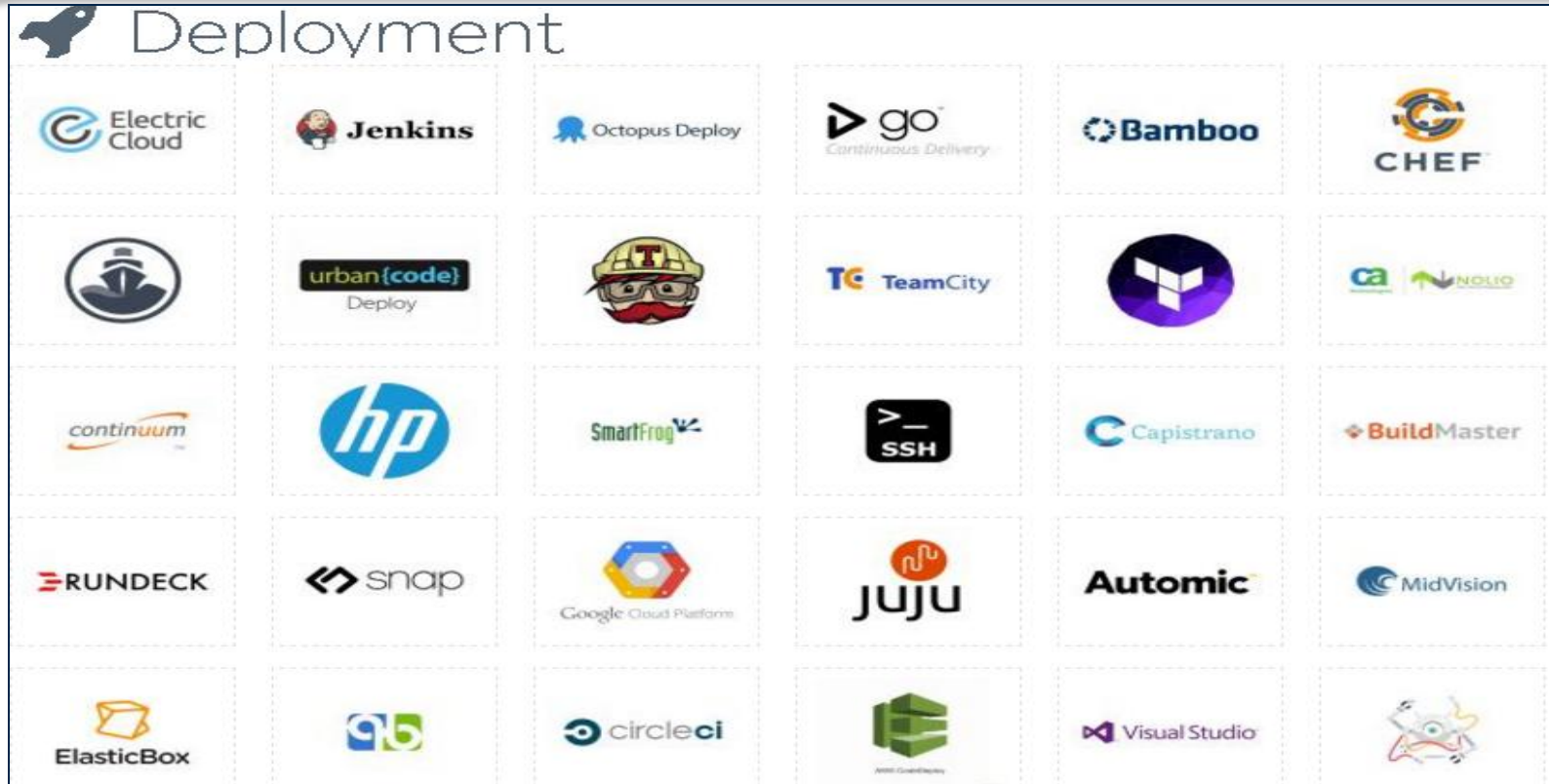
- Monitoring of network services (SMTP, POP3, HTTP, NNTP, ICMP, SNMP, FTP, SSH)
- Monitoring of host resources (processor load, disk usage, system logs)
- Monitoring of any hardware (like probes for temperature, alarms, etc.)
- Monitoring via remotely run scripts (via Nagios Remote Plugin Executor)
- Remote monitoring (supported through SSH or SSL encrypted tunnels)
- And many more monitoring features.



Continuous Deployment and Release Management

- Continuous deployment and release management raise the concept of continuous integration to the next level enabling creation of the delivery pipeline .
- This pipeline automates continuous deployment of software to QA environment, then to production in an efficient manner.
- Continuous release and deployment makes it possible to release new features to customers and users at the earliest possible..
- Correct selection of tooling and processes make up the core of DevOps to facilitate continuous integration, continuous release, and continuous deployment.

Deployment Automation Tools



Release Management Tools



Release Management



Lesson Summary

- DevOps – the business need
- What is DevOps?
- How does DevOps Work?
- DevOps Practices
 - Source Code Management
 - Automated testing
 - Containerization
 - Continuous integration
 - Configuration Management
 - Automated Monitoring
 - Continuous Deployment and Release Management

