

# DevOps



People matter, results count.

# Table Of Contents

## Part I – Introduction

- DevOps – the business need?
- What is DevOps?
- What DevOps is not?
- How does DevOps work?



## Part II – Implementation and Tools

- DevOps Practices
- Infrastructure as Code
- Learn a Tool - Chef and Puppet
- Source Code Management
- Learn a Tool - GIT
- Continuous Integration
- Learn a Tool - Jenkins



- Automated Testing
- Learn a Tool - Selenium
- Continuous Deployment
- Release Management



## Part III – Introduction

- DevOps Cloud
- Cloud and DevOps using BlueMix
- BlueMix DevOps – Example



## Part II – Demo

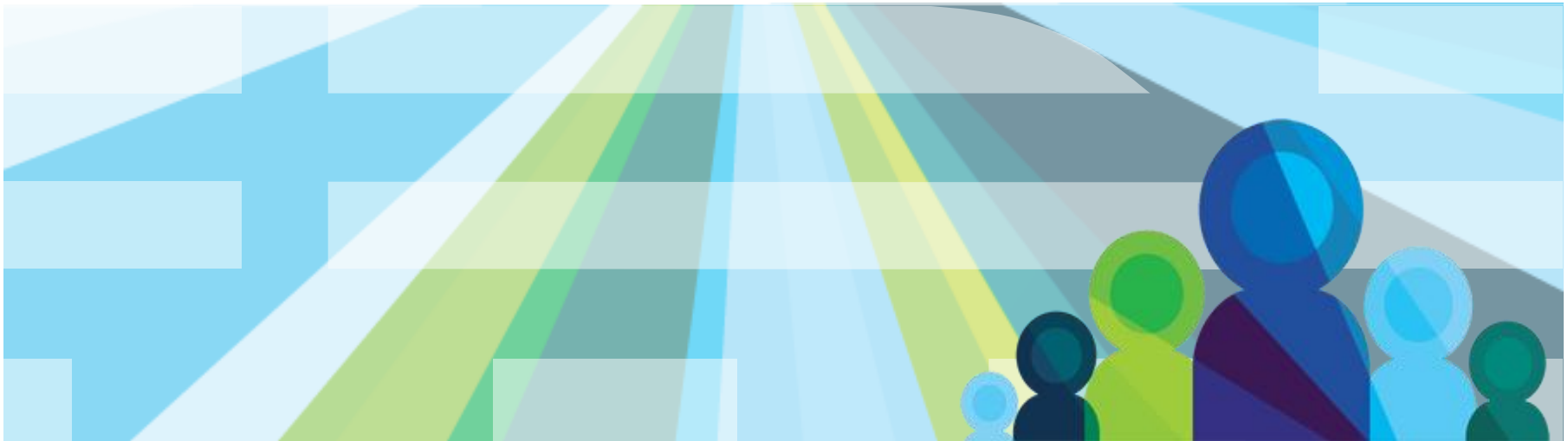
- DevOps Delivery Pipeline
- DevOps App Monitoring & Scaling
- DevOps Web Editor





---

# DevOps- Introduction



# DevOps – the business need



*The Developer*



**Time to Market**



**Dependency Error**

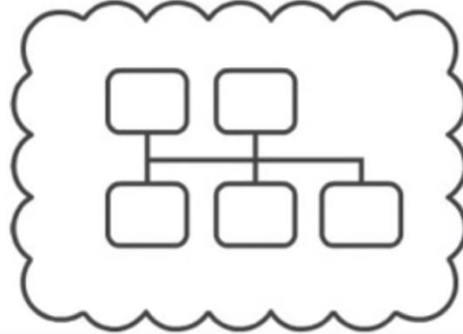
# DevOps – the business need



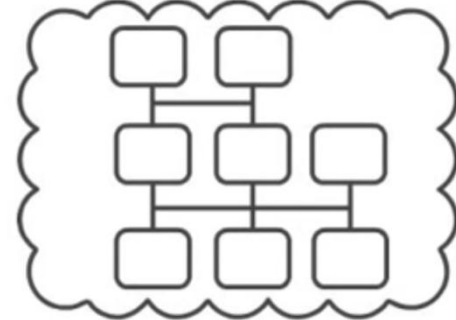
*The Developer*

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**



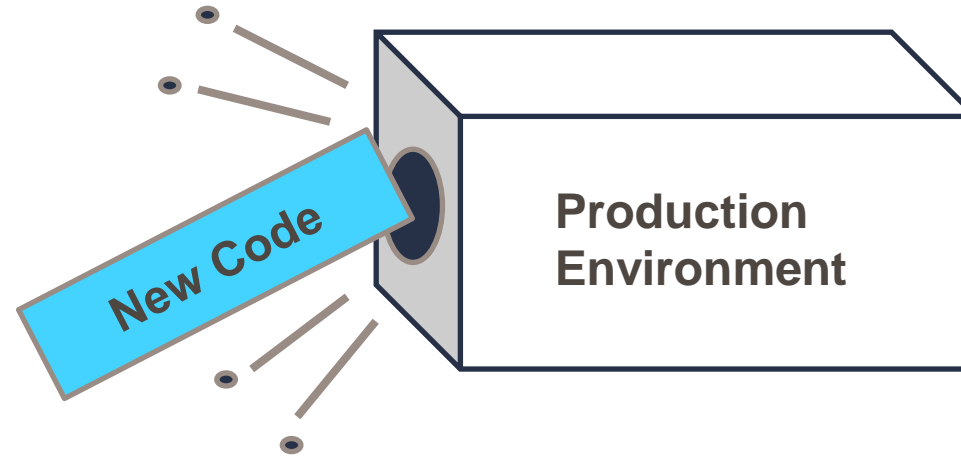
**PRODUCTION  
ENVIRONMENT**



# DevOps – the business need



*The Operations team*



Deployment Schedule						
Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday

# 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*





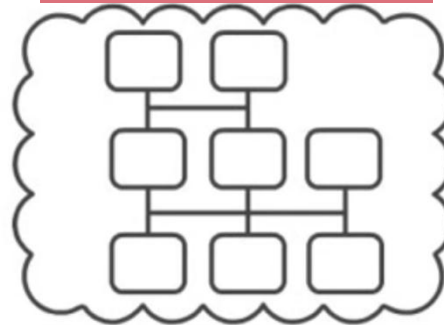
# DevOps – the business need

*DevOps*

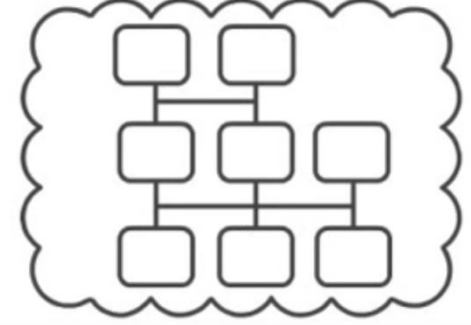


- ✓ **Worked Better together**
- ✓ **Thought more alike**
- ✓ **Broke down silos**
- ✓ **Shared responsibilities?**

**DEVELOPMENT  
ENVIROMNEMT**



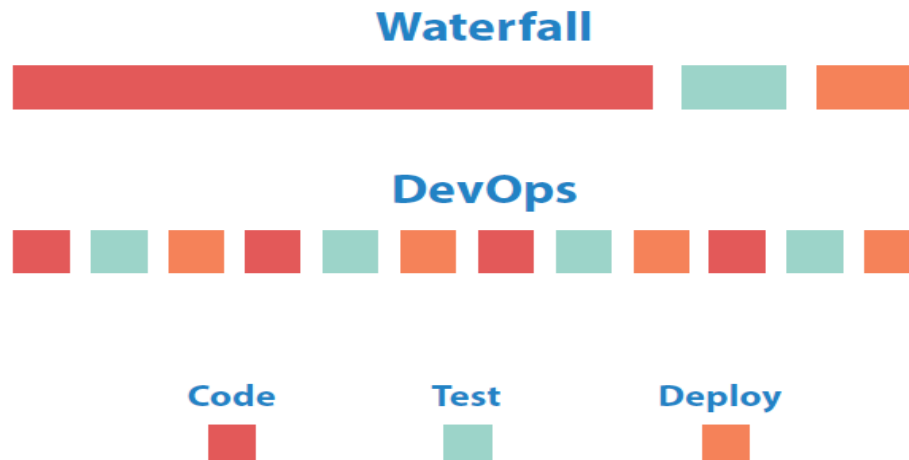
**PRODUCTION  
ENVIROMNEMT**



# 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

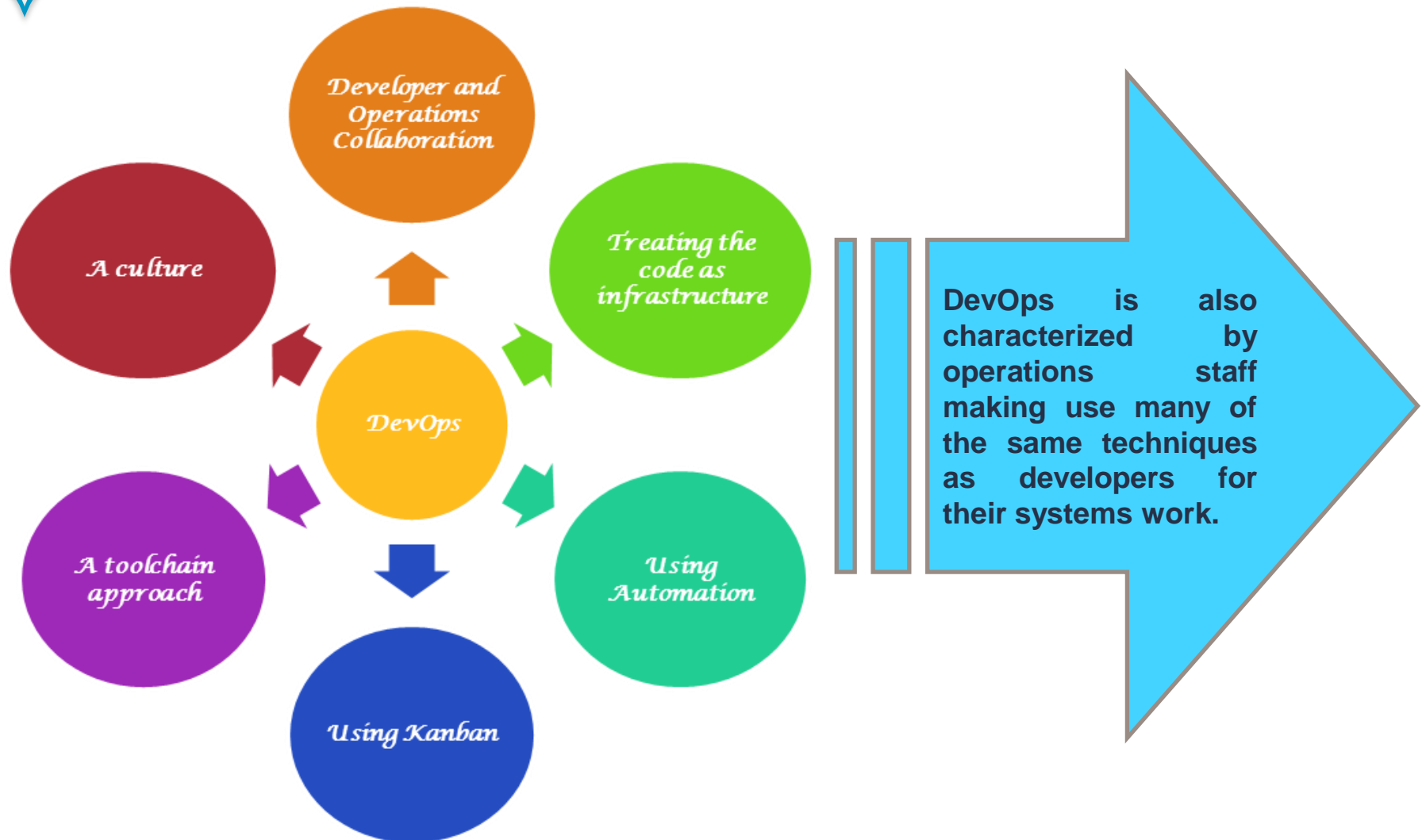
**A** – Automation

**L** – Lean

**M** – Measurement

**S** – Sharing

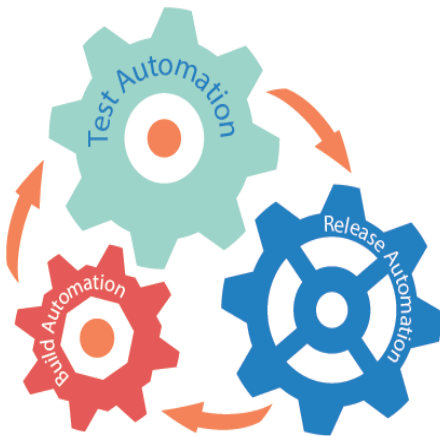
# What is DevOps?



# What is DevOps?

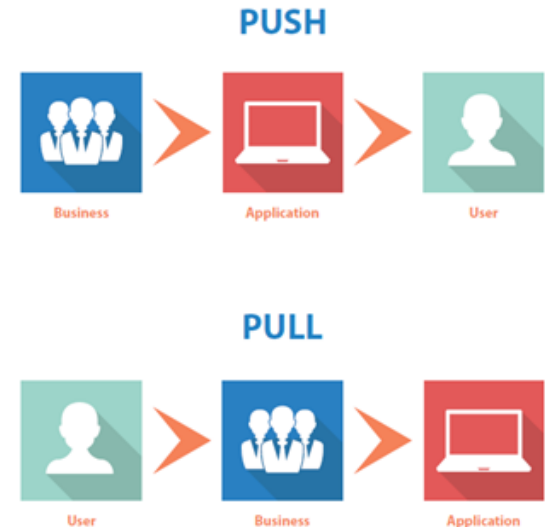
## *Automation - Optimizing the Entire Pipeline*

- ✓ 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, and automation is key to realizing this goal.



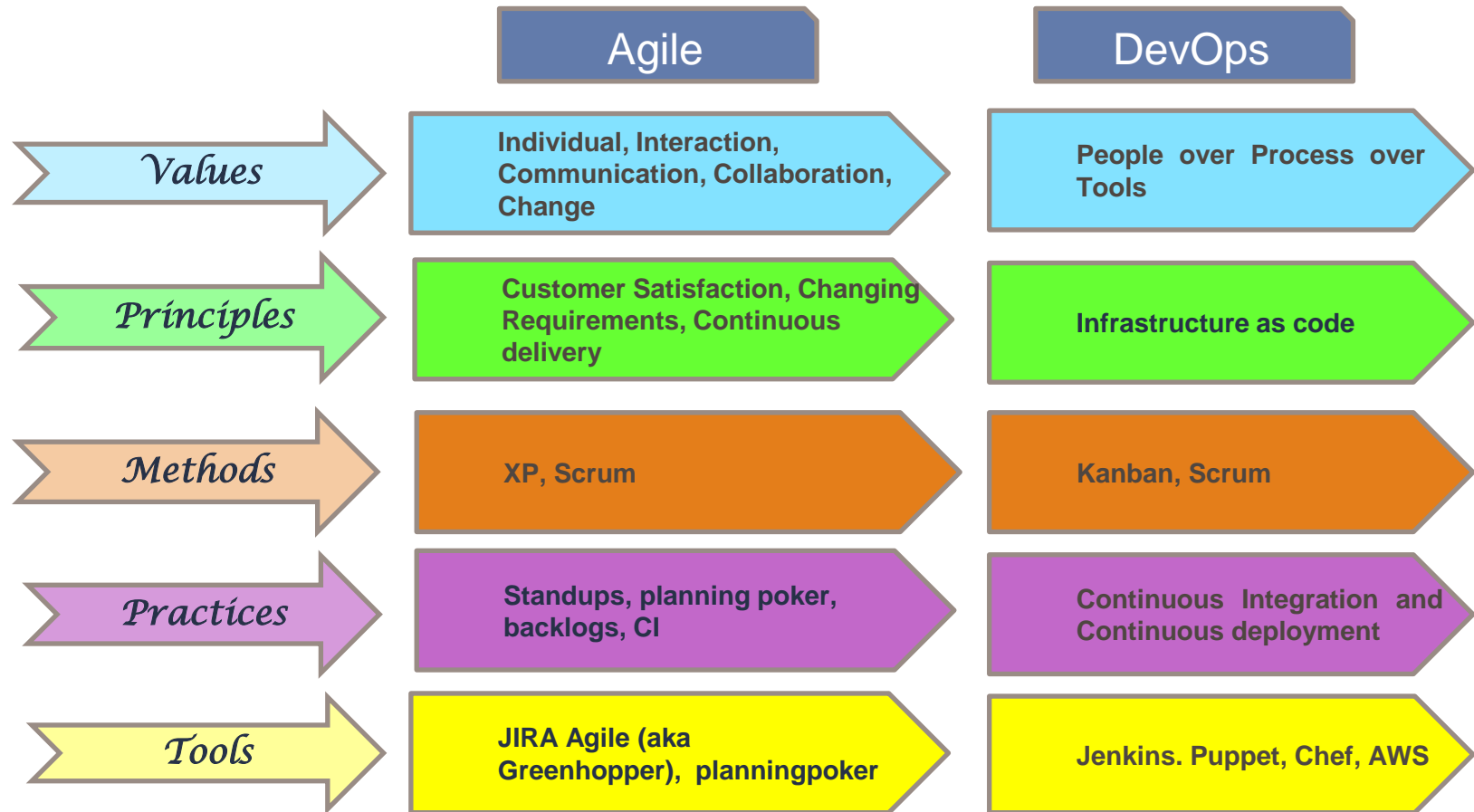
## **PUSH vs PULL**

- ✓ 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 is DevOps?

## *Agile and DevOps - A parallel definition*



# 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!

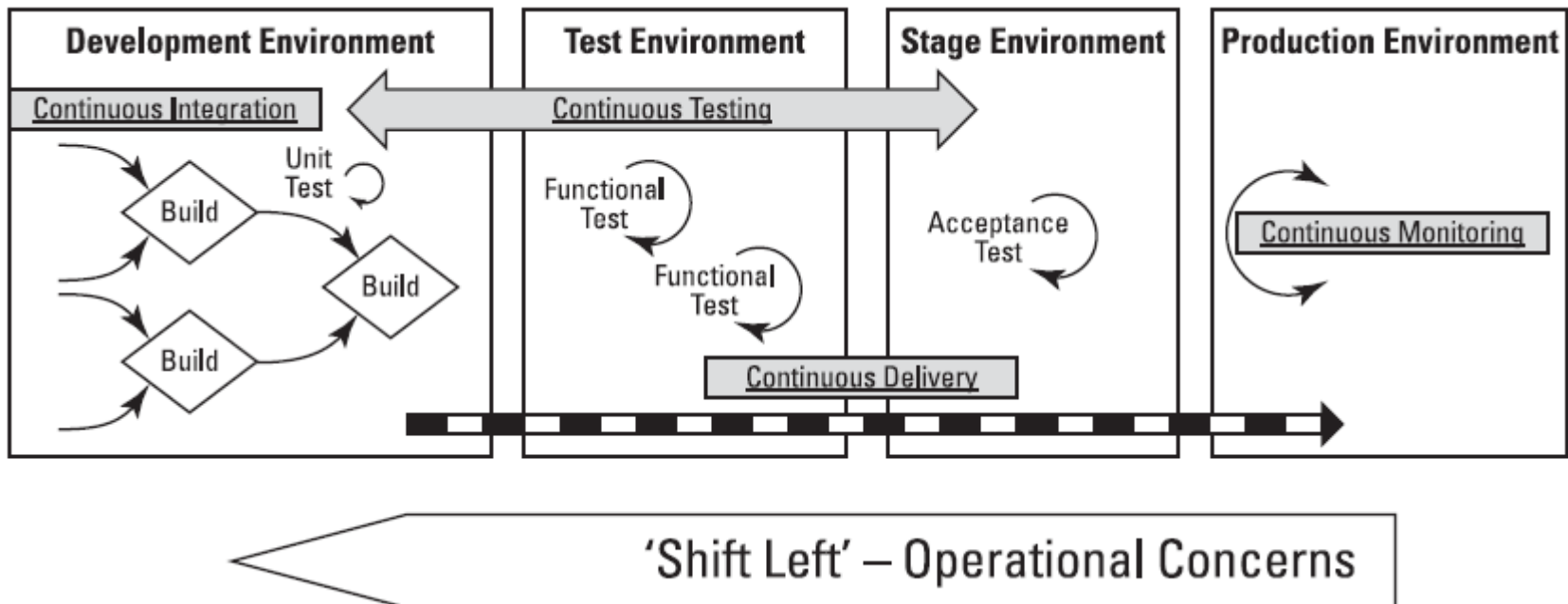
- **It's Not Everything**

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?

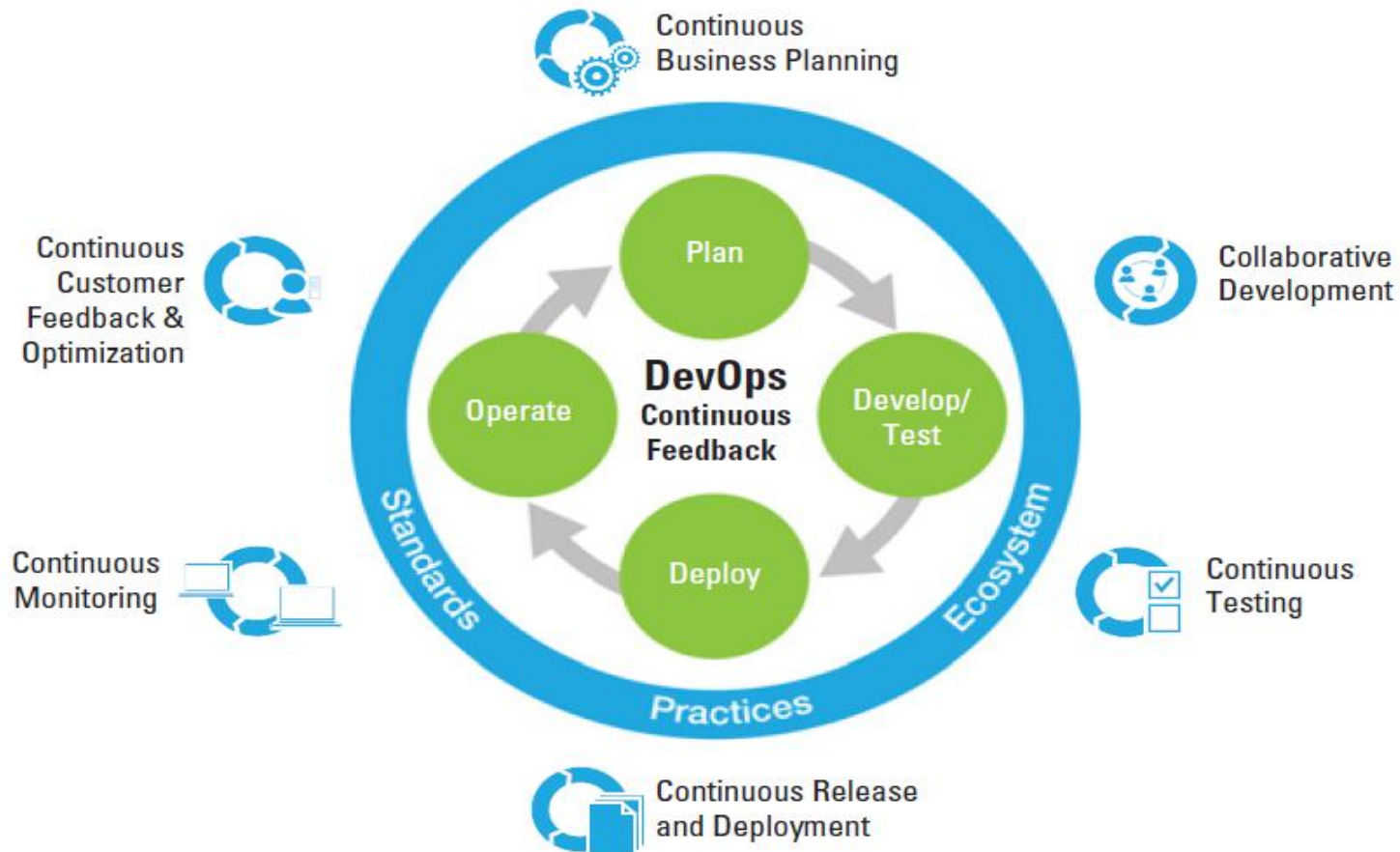
- **The 4 principles:**

- ✓ 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:**

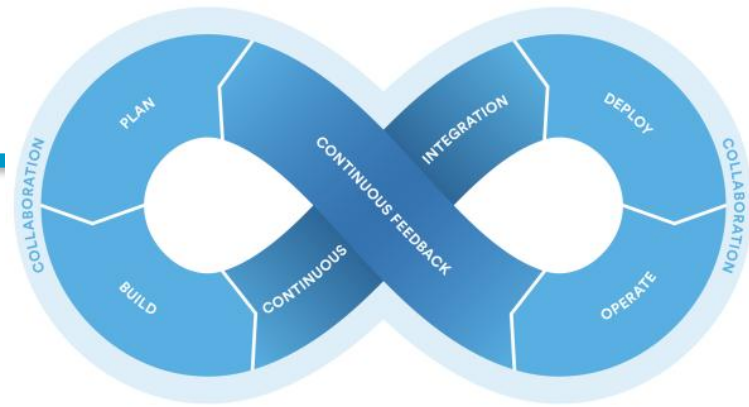
Forms the core of development and quality assurance (QA) capabilities. It involves two practices - collaborative development and continuous testing.

- **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 feedback.



# How does DevOps Work?

No tool will magically make the team DevOps.

## Develop



git  
Subversion  
PERFORCE  
Version everything.  
Atlassian  
Bitbucket

## Test



Jenkins  
maven  
gradle  
Se

## Deploy



Capistrano  
Jenkins  
Visual Studio  
Team Foundation Server

## Monitor



Nagios  
New Relic  
QUALYS  
Ganglia  
icinga  
pagerduty  
sensu  
cloudMAP  
Lantana

## Log



GRAYLOG2  
Open source Log Management  
papertrail  
logstash  
loggly  
splunk>  
sumologic  
Upstart  
cloudMAP

## Configuration Management



puppet labs  
CHEF  
ANSIBLE  
docker  
CFEngine  
VAGRANT  
cloudMAP

## Security



threat stack  
tripwire  
cloudMAP

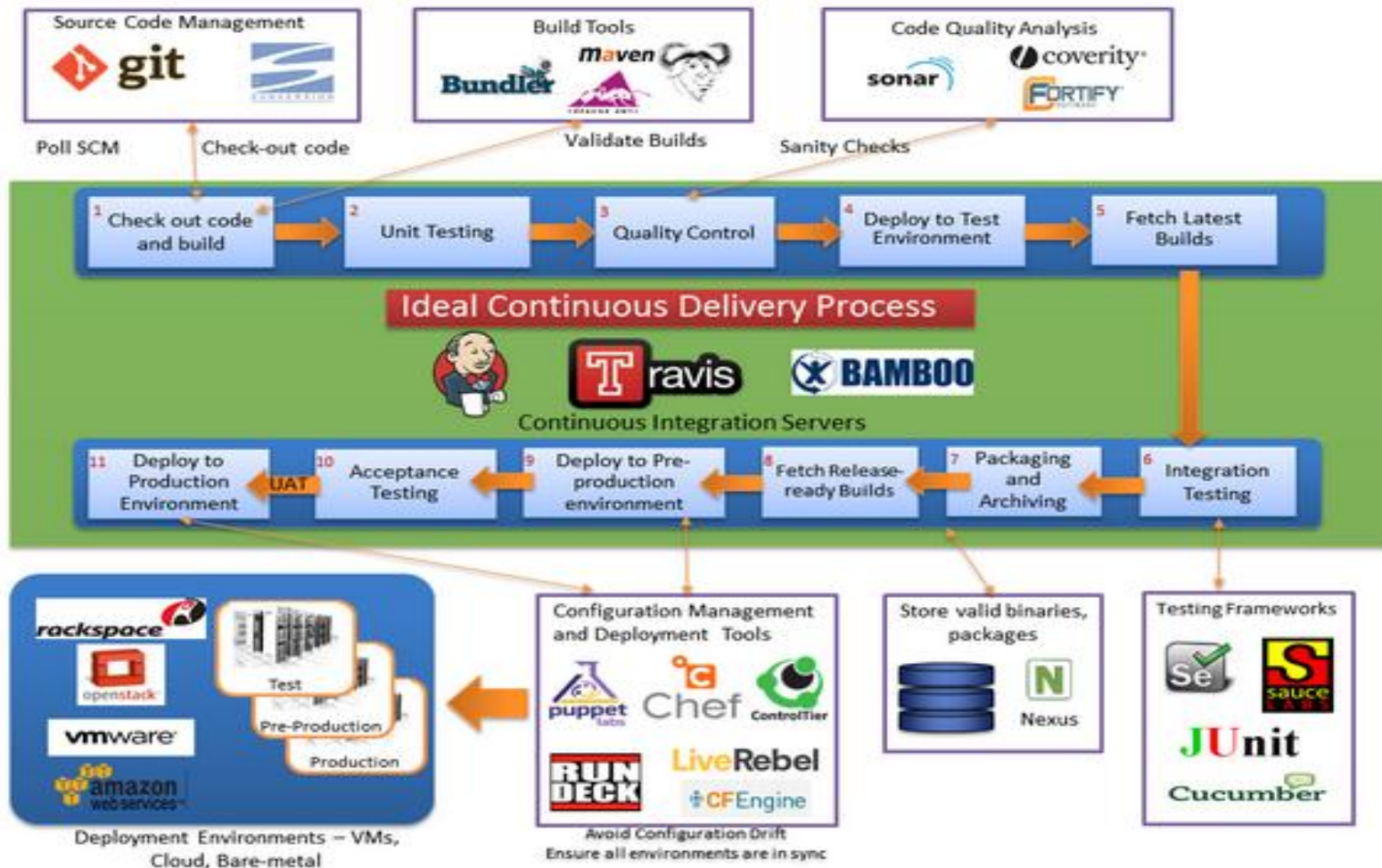
## Collaboration Platform



slack  
Trello  
RALLY  
Visual Studio  
Team Foundation Server  
Whizzible  
ITinvolve

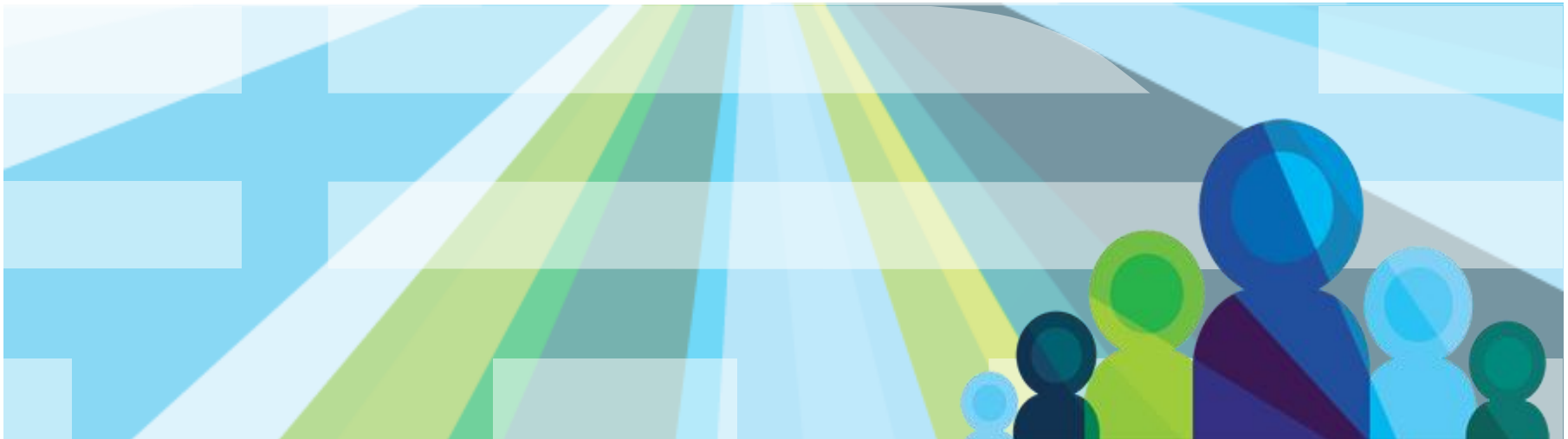
# How does DevOps Work?

## *The Continuous Delivery Pipeline*



---

# DevOps- Implementation and Tools



# DevOps Practices

## *DevOps Practices:*

- Infrastructure as Code (IaC)
- Source Code Management
- Continuous Integration
- Automated Testing
- Continuous Deployment
- Release Management

# Infrastructure as Code (IaC)

*IaC is not a product, it's a methodology*

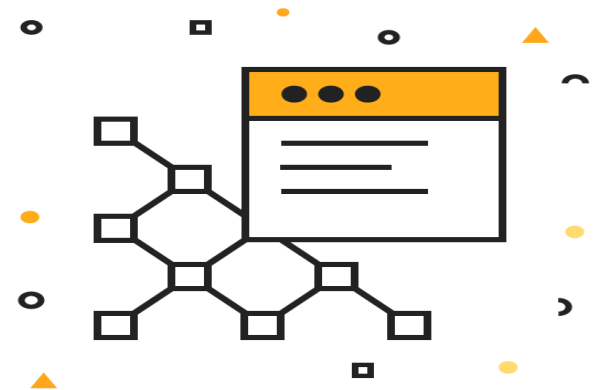
## *Infrastructure as Code (IaC)*

- Organizations looking for faster deployments treat infrastructure like software
- Infra as code that can be managed with the same tools and processes software developers use, such as version control, continuous integration, code review and automated testing.
- Makes infrastructure changes more easy, rapid, safe and reliable.

### **To properly embrace IaC, you need three things:**

- agile development processes
- a DevOps environment
- the tools to write the code.

### **Example : Chef / Puppet**

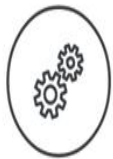
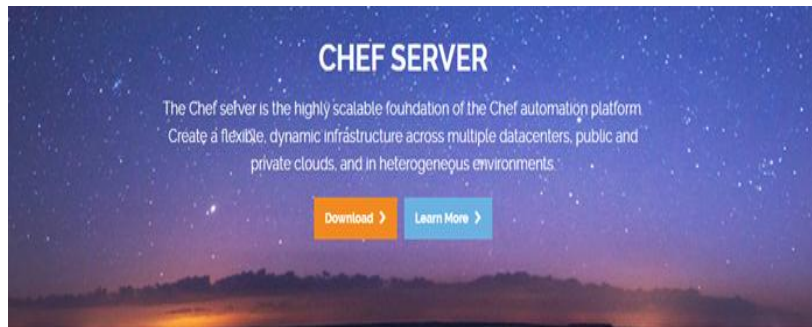




# Learn the Tool – Chef and Puppet

## *Infrastructure as Code (IaC) Tool*

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



### 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

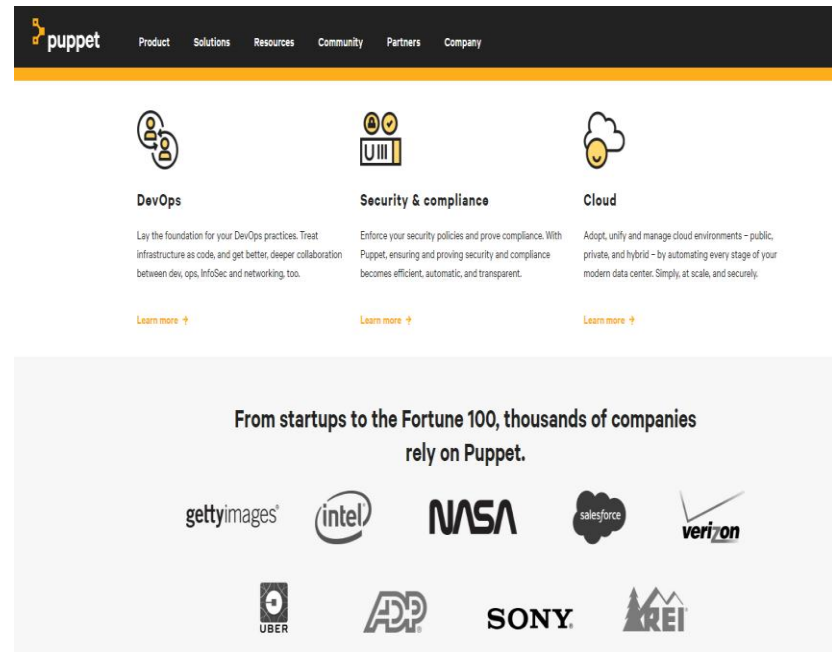
Automatically deliver the latest tested and approved policies to your infrastructure.



### Scale

Manage complexity with the most scalable automation platform on the planet.

<https://puppet.com>



# Learn the Tool – Chef

## 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`.

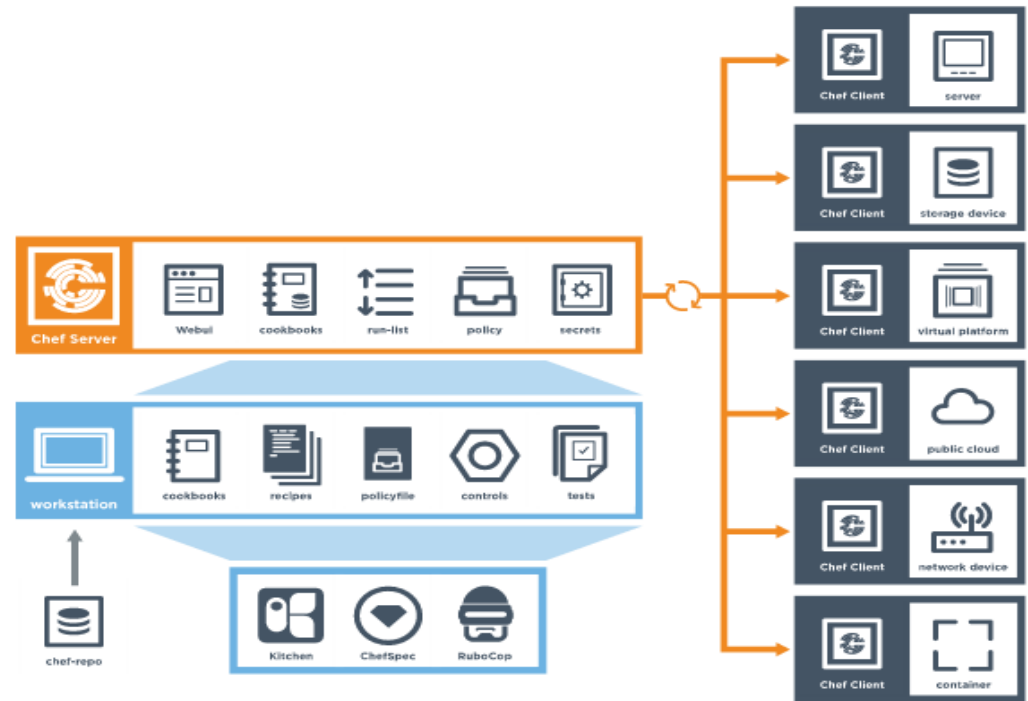
```
Editor: ~\chef-repo\webserver.rb
1 powershell_script 'Install IIS' do
2   code 'Add-WindowsFeature Web-Server'
3   guard_interpreter :powershell_script
4   not_if "(Get-WindowsFeature -Name Web-Server).Installed"
5 end
```



# Learn the Tool – 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

# Learn the Tool – Puppet

## *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

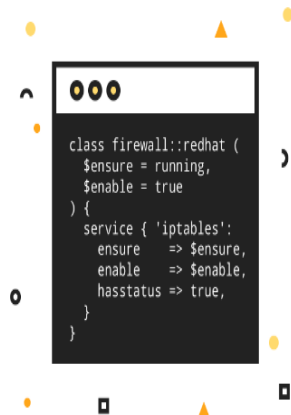
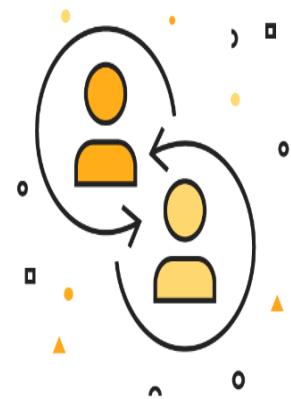
Automation — the foundation for many DevOps practices — helps you move faster without sacrificing stability or security. Now is the time to take advantage of automation and proven DevOps practices to drive your team — and your deployments — forward.

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.





# Source Code Management

## *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**

# Learn the Tool – GIT

## Source Code Management Tool

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



The screenshot shows the Git website homepage. At the top left is the Git logo and the tagline "--distributed-even-if-your-workflow-isnt". To the right is a search bar. Below the tagline, there are two paragraphs of text describing Git as a free and open source distributed version control system, highlighting its speed, efficiency, ease of learning, and performance. A diagram illustrates the distributed nature of Git with multiple stacks of code representing repositories connected by lines. Below the text, there's a section titled "Learn Git in your browser for free with Try Git." with a small Git logo. At the bottom, there are four icons with corresponding text: "About" (gears icon), "Documentation" (book icon), "Downloads" (download arrow icon), and "Community" (speech bubble icon). On the right side of the bottom section, there's a monitor displaying the "Latest source Release 2.9.3" and 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.

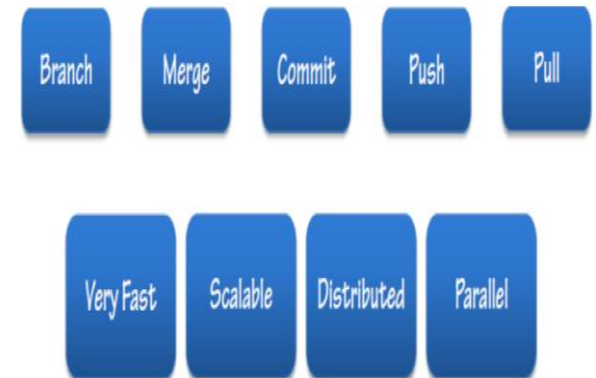
**Documentation**  
Command reference pages, Pro Git book content, videos and other material.

**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

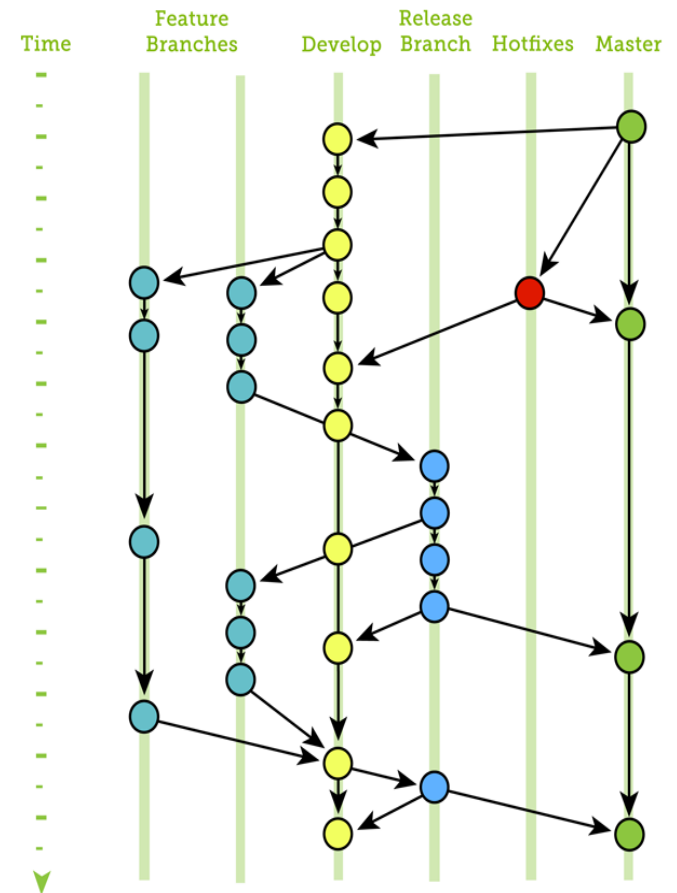
Distributed Version  
Control System  
("DVCS")



# Learn the Tool – 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.



# Continuous Integration (CI)



Fail Fast

## *Continuous Integration*

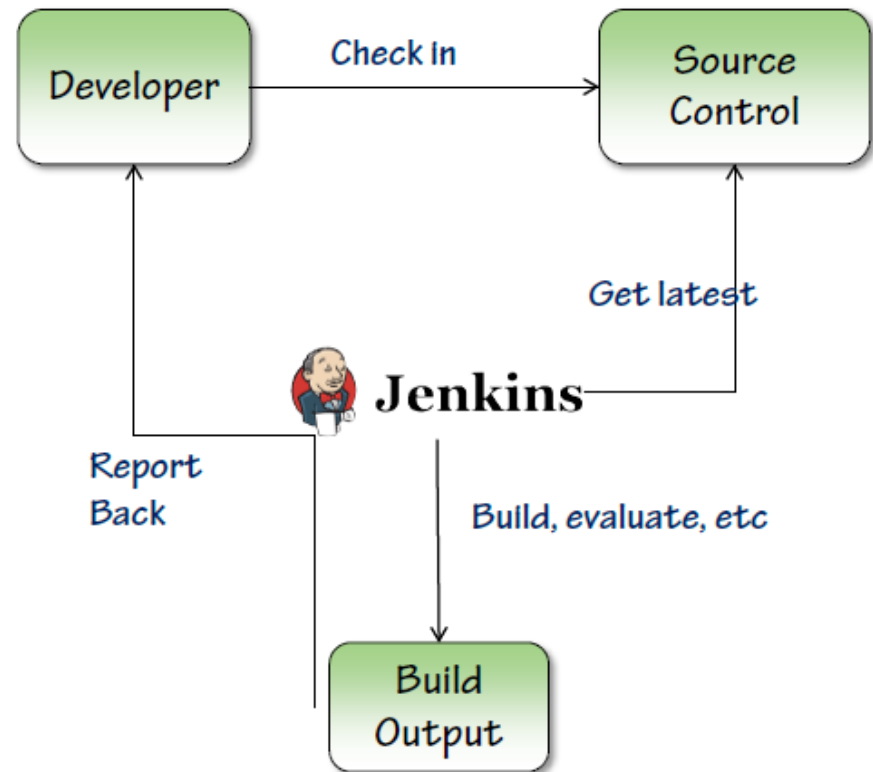
- Continually merges source code updates from all developers on a team into a shared mainline.
- Prevents a developer's local copy of a software project from drifting too far afield as new code is added by others, avoiding catastrophic merge conflicts.
- CI involves a centralized server that continually 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**

# Learn the Tool – Jenkins

## *Continuous Integration (CI) Tool*

<https://jenkins.io>

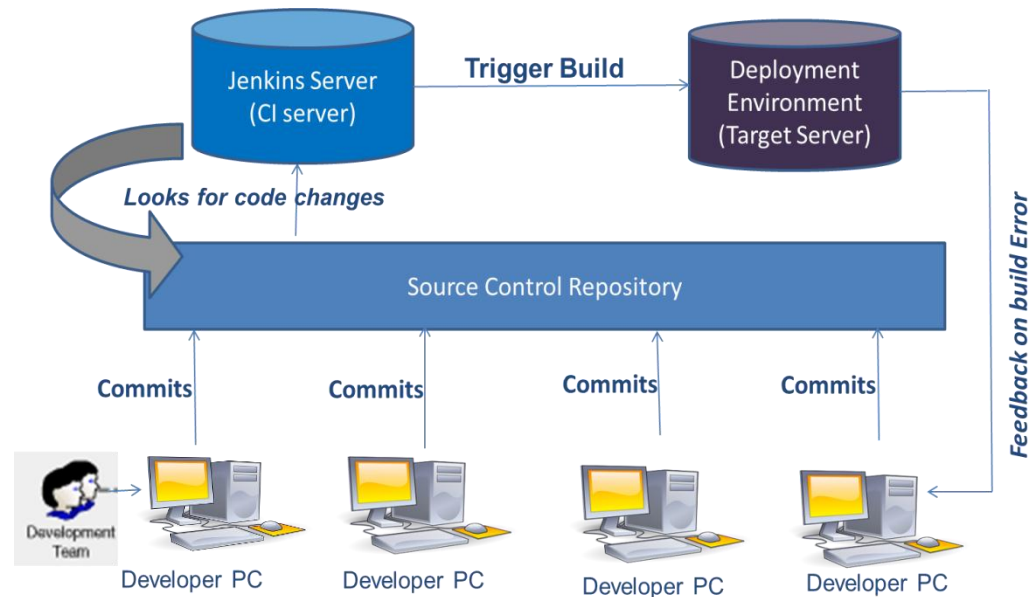




# Learn the Tool – Jenkins

## 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

# Automated Testing

## *Automated Testing*

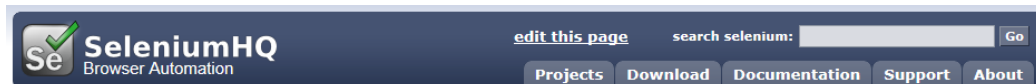
- The objective of automated testing is to simplify as much of the testing effort as possible with a minimum set of scripts.
- Automated testing tools are capable of executing repeatable tests, reporting outcomes, and comparing results with faster feedback to the team.
- Automated tests perform precisely 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

## **Example : Selenium**

# Learn the Tool – Selenium

## Automated Testing Tool

<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

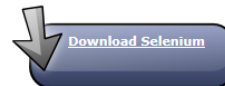
- create quick bug reproduction scripts
- create scripts to aid in automation-aided exploratory testing



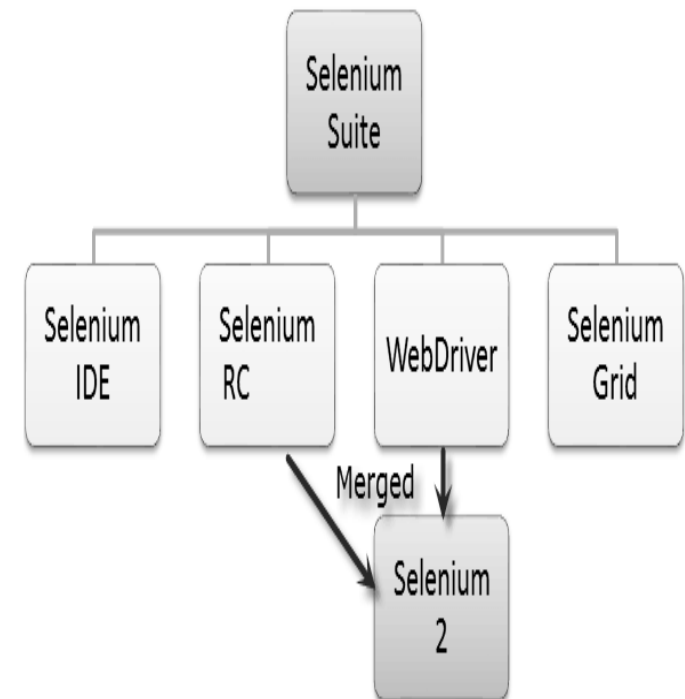
**Selenium is a suite of tools** to automate web browsers across many platforms.

Selenium...

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



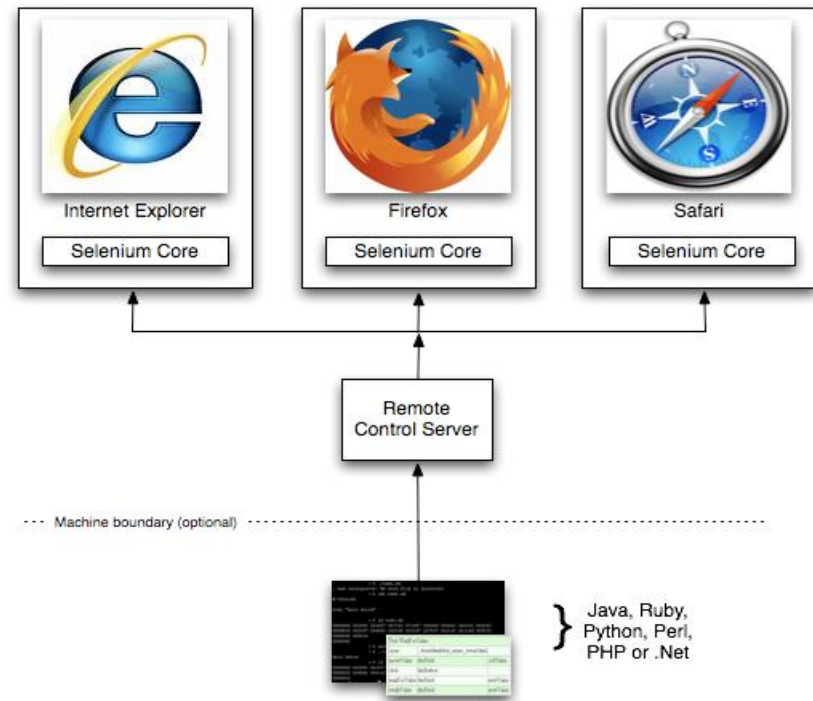
[Donate to Selenium](#)



# Learn the Tool – 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.



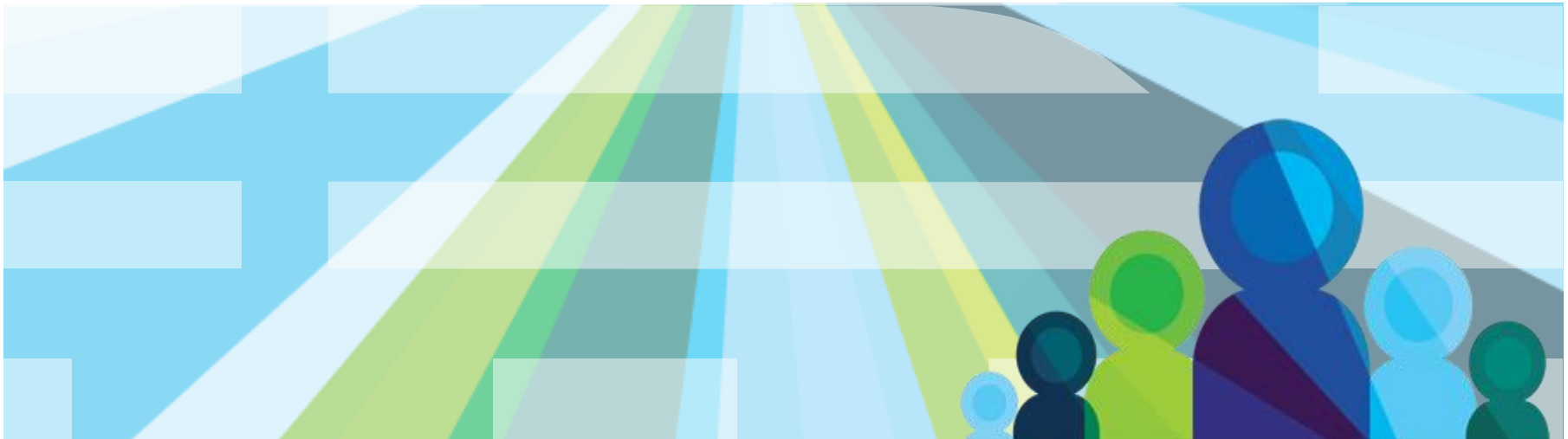
# Continuous Deployment and Release Management

## *Continuous Deployment & 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.

---

# DevOps – DevOps Cloud



# Cloud and DevOps

## *Cloud and DevOps:*



- Cloud and DevOps are independent but mutually reinforcing strategies for delivering business value through IT.
- The principle of continuous improvement is key to Agile, cloud, and DevOps.
- Cloud providers are more than happy to support their customers' DevOps needs.
- Nearly every major cloud provider offers a set of platform as a service (PaaS) tools that are fine-tuned to their environment.



*Like Chocolate & Peanut Butter*



### *Automated Application Deployment*

- ✓ Automate applications deployments to reduce errors
- ✓ Quickly release and deploy apps to 1 or 1000's of servers
- ✓ Manage complex environments

### *Faster Application Development*

- ✓ Continuously deliver apps for rapid innovation
- ✓ Reduce capital costs by avoiding infrastructure purchases
- ✓ Access metrics and customer feedback

### *DevOps for Monitoring*

- ✓ Manage the performance, availability and capacity of your apps
- ✓ When an app fails, quickly understand why



## *Cloud and DevOps - Principles:*

- DevOps practice in cloud organization uses disciplines such as agile software delivery methodologies and building tools and techniques to support continuous delivery and deployment.

## *Solutions : IBM BlueMix, AWS*

### *Principles:*

- **Everything is code.** In the cloud, everything must be treated as code.
- **Life is too short for bad software, so find ways to improve it.** If you can't measure it, you can't fix it.
- **Stop running things manually.** Catching up automation with continuous delivery in the cloud is not an easy task to accomplish if you have not developed a culture of automation.
- **Know before your customer knows.** The real benefit of monitoring and alerting infrastructure comes when you know about a problem before your users see it.





What would you do when u want to  
have Pizza?





## Pizza as a Service



☒ You Manage    ☐ Vendor Manages



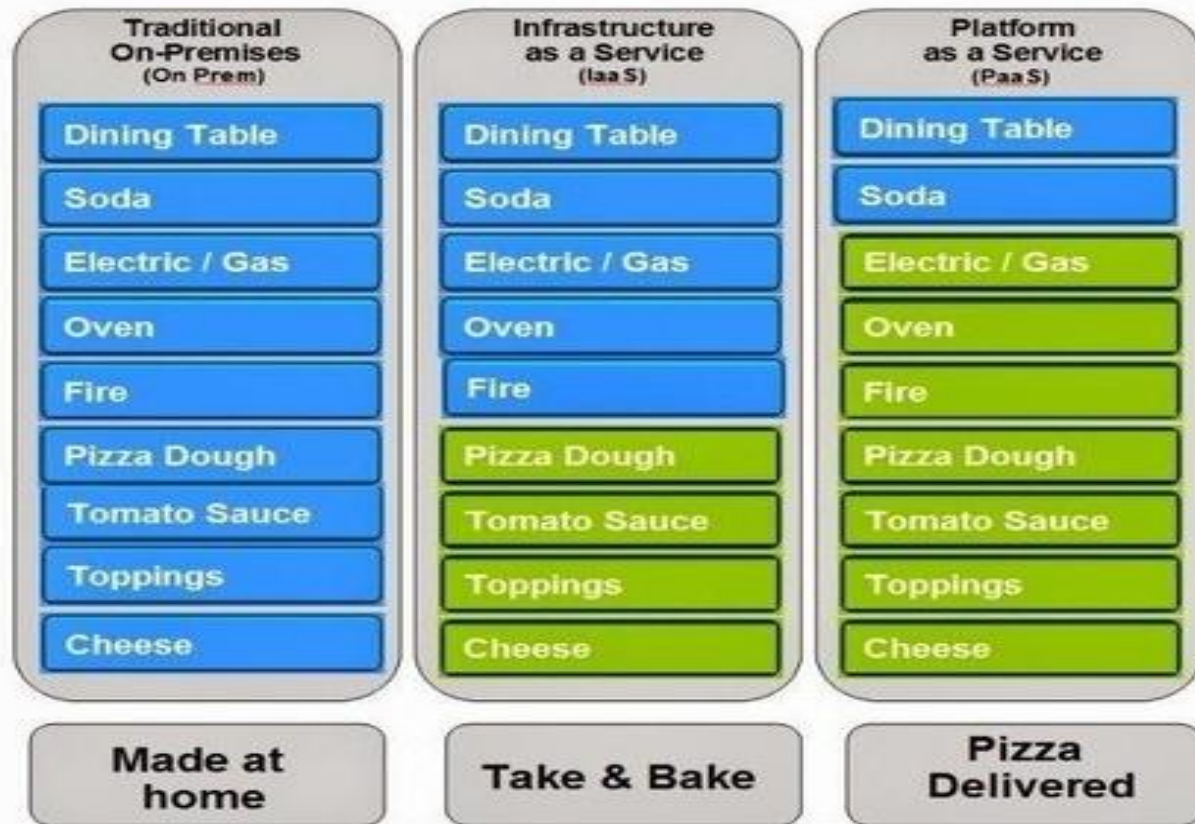
## Pizza as a Service



■ You Manage ■ Vendor Manages



## Pizza as a Service



■ You Manage ■ Vendor Manages





## Pizza as a Service

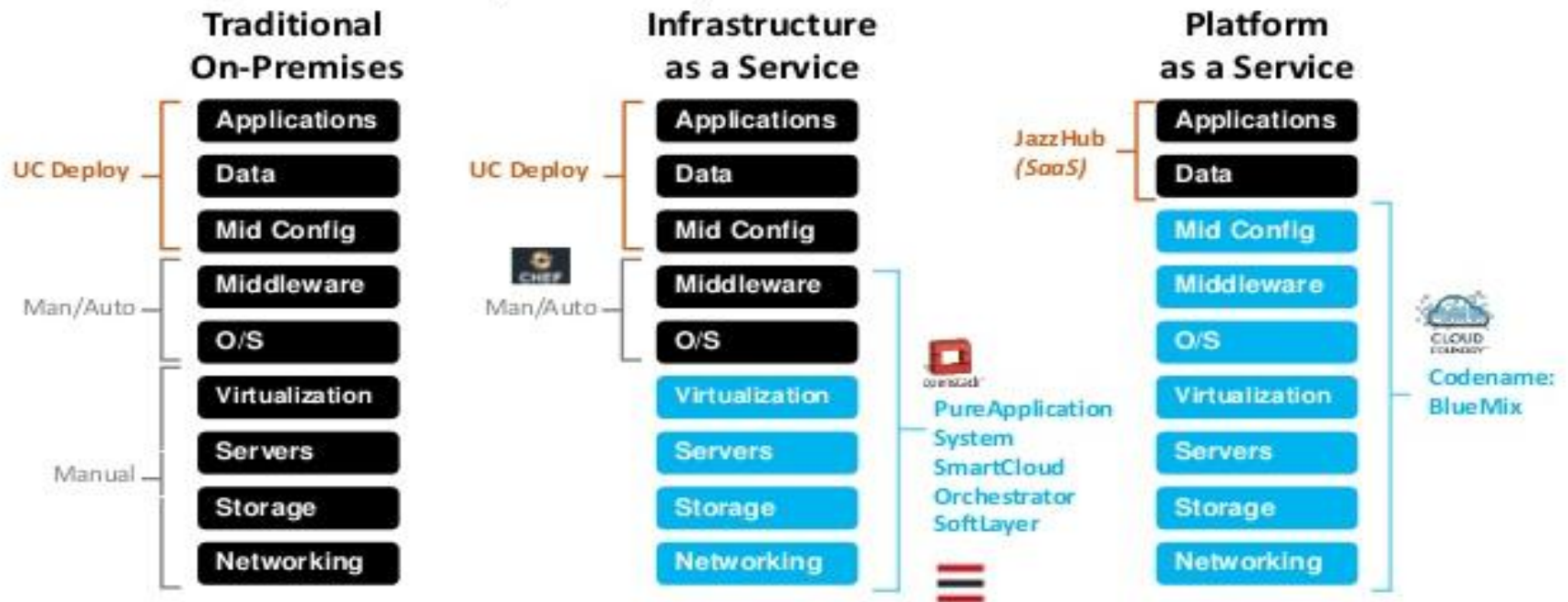


# Cloud and DevOps



## DevOps and Cloud adoption

*Automating for faster delivery with DevOps and cloud*



Customization; higher costs; slower time to value

Standardization; lower costs; faster time to value

# Cloud and DevOps using BlueMix



<https://hub.jazz.net/>

## DevOps Made Easy

Integrated Agile Planning, Coding, Building, Deploying

Sign up for free

We bring the tools. You bring the code.



Easy Access

Get started for free. With Git hosting and the built-in Web IDE, it's zero to code in seconds.



Code Now

Use the built-in Web IDE, Eclipse, Visual Studio, or your tool of choice.



Build & Deploy

Automatically build and deploy your application to IBM's cloud platform, Bluemix.



Team Collaboration

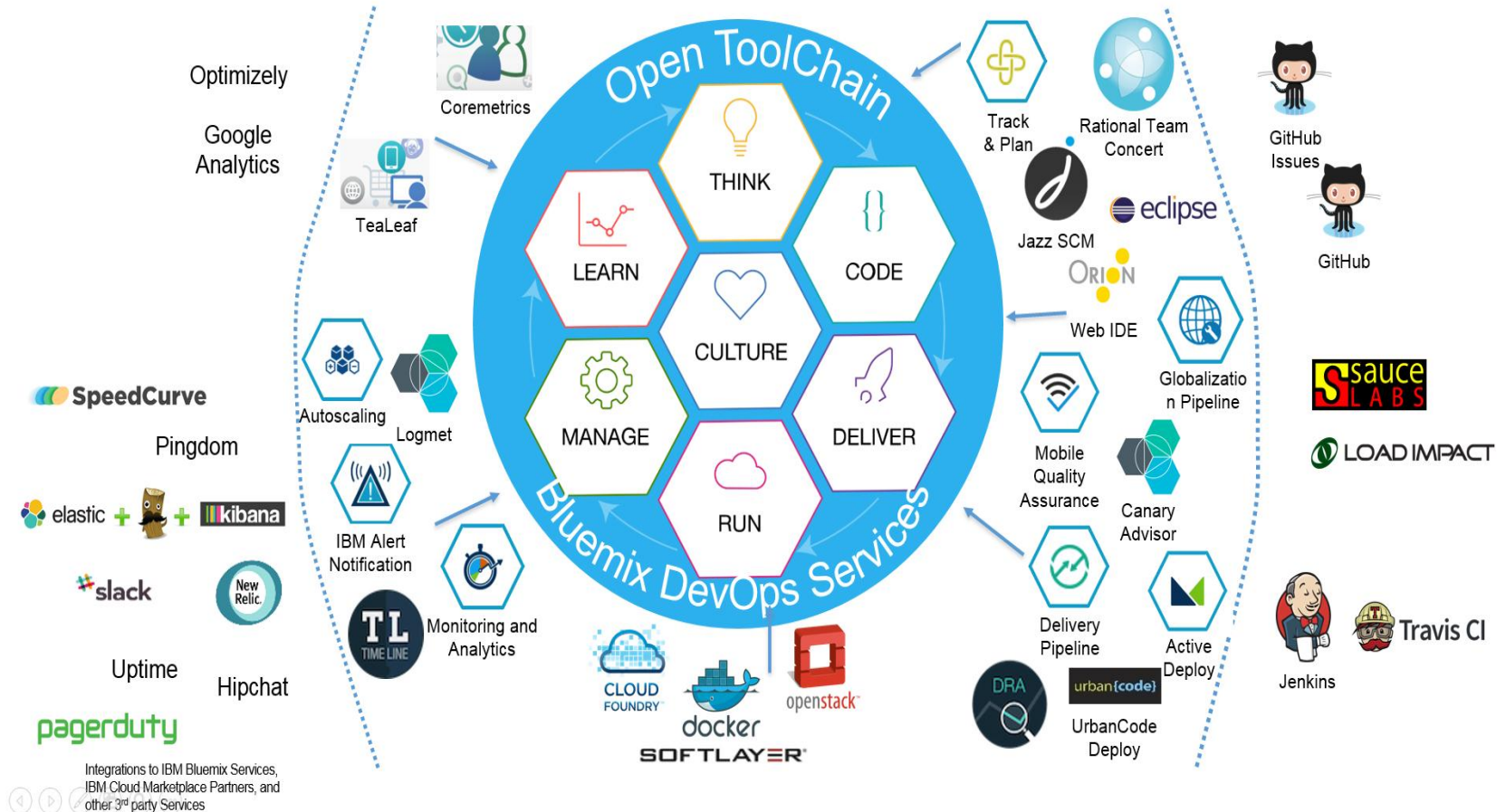
Share your work and collaborate through expert tools for Agile Development.

## *IBM Bluemix DevOps Services :*

- Git hosting
- Connect with your GitHub repository
- Work in one place – the cloud
- Simple user interface for starting a project quickly
- No installation, just code
- Integrated source code editor
- Update a running app automatically
- Built-in source code management
- Build and deploy
- Work items to track and plan project activities
- Dashboard charts for project status

# Cloud and DevOps using BlueMix

## DevOps Toolchain – from IBM BlueMix Page





# BlueMix DevOps - Example



## Create a Project:

Fork Project

Name your project:

ibmdevopsservices | Project

✓

URL: <https://hub.jazz.net/project/ibmdevopsservices/Project>

☒ Make it private (not public)

☒ Add features for Scrum development ⓘ

☒ Make this a Bluemix Project ⓘ

Select a Bluemix space to bill your services to:

Region

IBM Bluemix (US South) ▼

Organization

uateam@ca.ibm.com ▼

Space

dev ▼

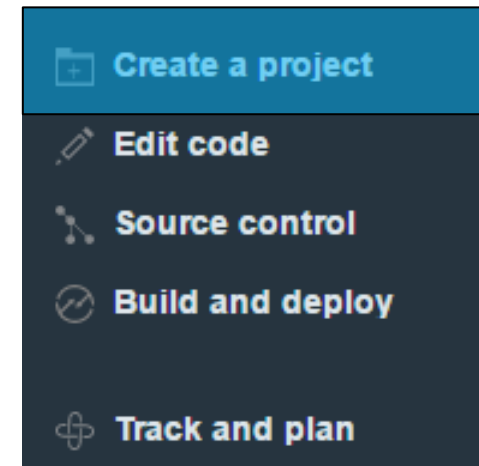
These selections can be changed later in the options for your Project Settings.

CANCEL

CREATE

## 2 ways:

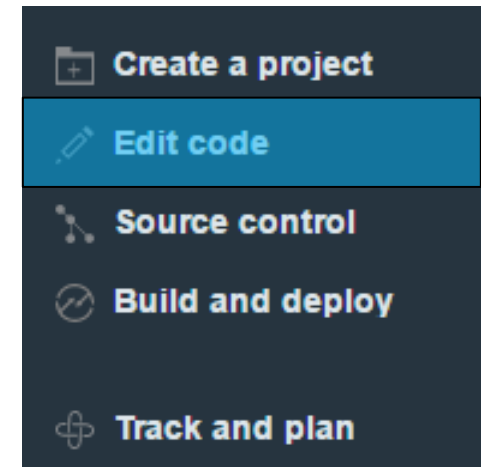
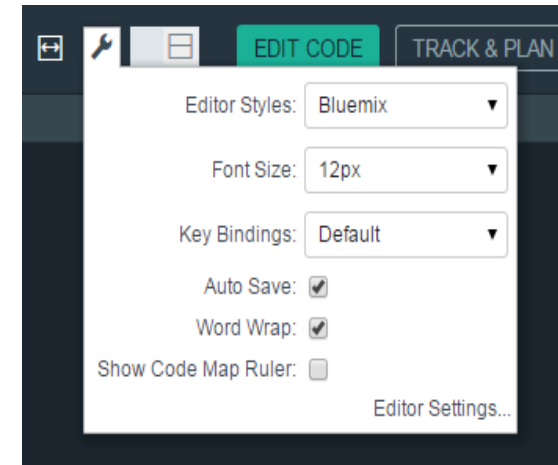
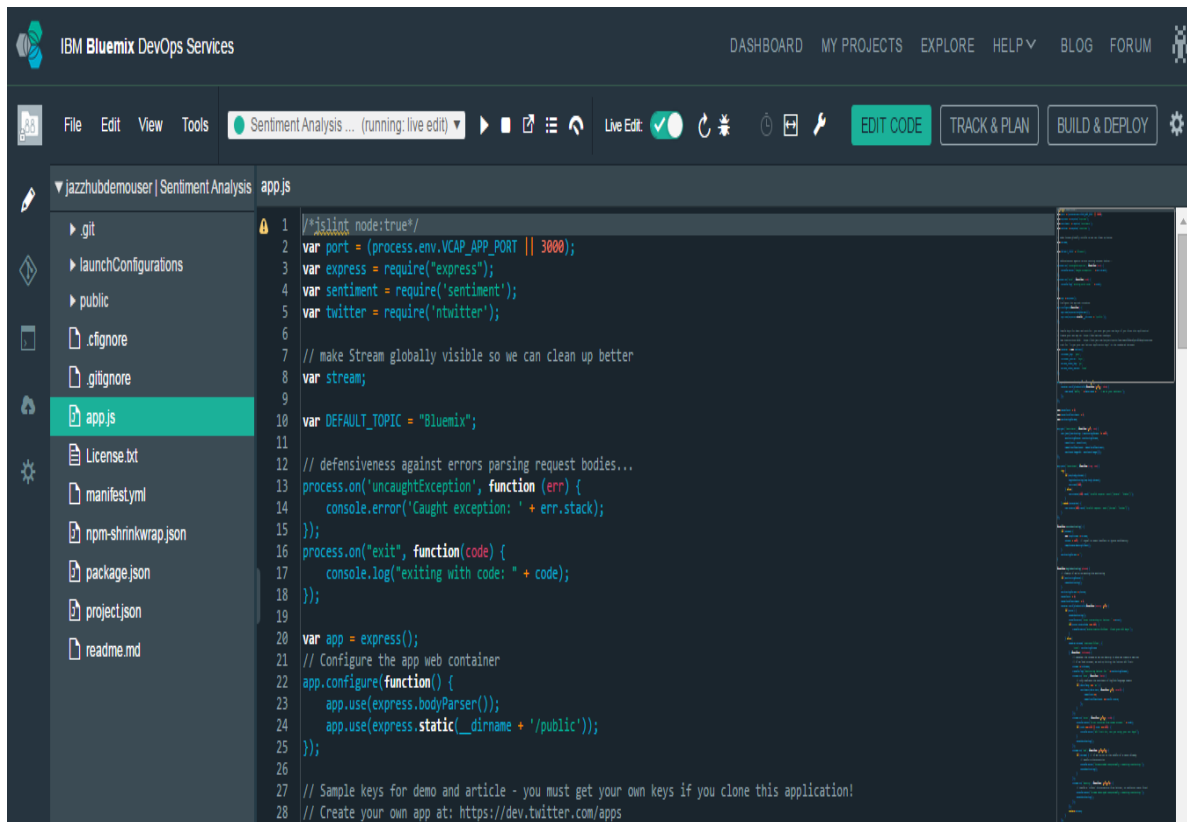
- A new project can be created
- Existing project can be Forked



# BlueMix DevOps - Example



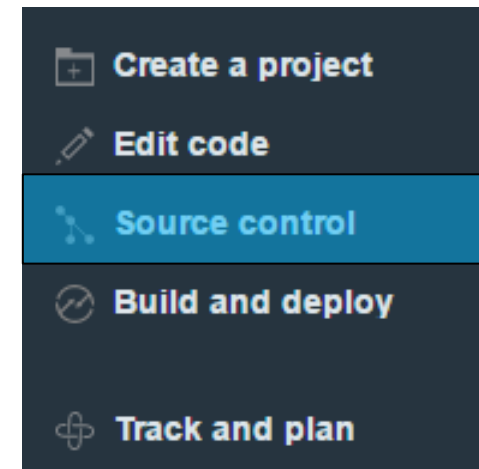
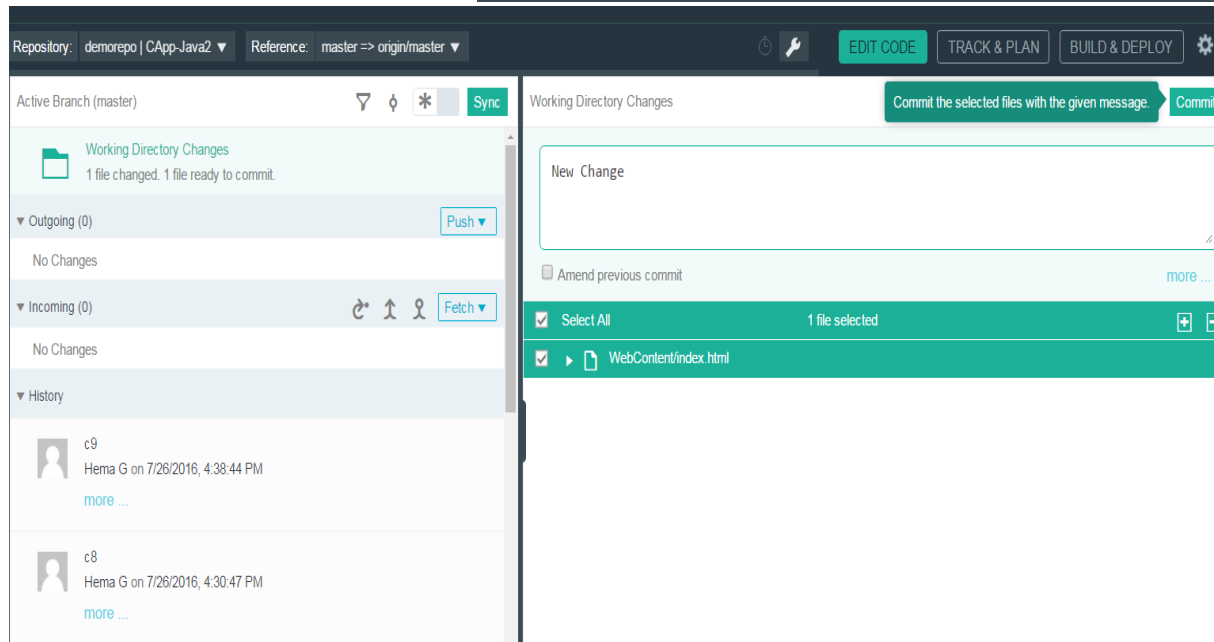
## Edit Code:



# BlueMix DevOps - Example



## Source Code:



# BlueMix DevOps - Example



## *Build & Deploy:*

Pipeline: All Stages

### Build Stage

STAGE PASSED

LAST INPUT [Git URL](#)

Last commit by Hema G [c9](#) 23 d ago

**JOBS** [View logs and history](#)

Build Succeeded 23 d ago

LAST EXECUTION RESULT

Build 9

### Deploy Stage

STAGE PASSED

LAST INPUT Stage: Build Stage / Job: B...

Build 9

**JOBS** [View logs and history](#)

Deploy to dev Succeeded 23 d ago

LAST EXECUTION RESULT

CApp-Java2 [CApp-Java2.au-syd.mybluemix.net](#) [View runtime log](#)

Build 9

+ ADD STAGE

**Create a project**

**Edit code**

**Source control**

**Build and deploy**

**Track and plan**

# BlueMix DevOps - Example



## *Track & Plan:*

Diagram illustrating the workflow for creating a work item in BlueMix DevOps:

- Add a brief description
- Add a detailed description
- Add subscribers, who will be notified of changes to the work item
- Set the work item's due date
- Set the work item's priority

Form fields and actions:

- Type a work item summary
- CREATE
- CANCEL

Diagram illustrating the workflow for creating a work item in BlueMix DevOps (continued):

- Select the work item type
- Select the team to file the work item against
- Set the work item's owner
- Add a tag of your choice to the work item
- Set the work item's parent

Form fields and actions:

- Create a project
- Edit code
- Source control
- Build and deploy
- Track and plan

---

# DevOps – Demo

