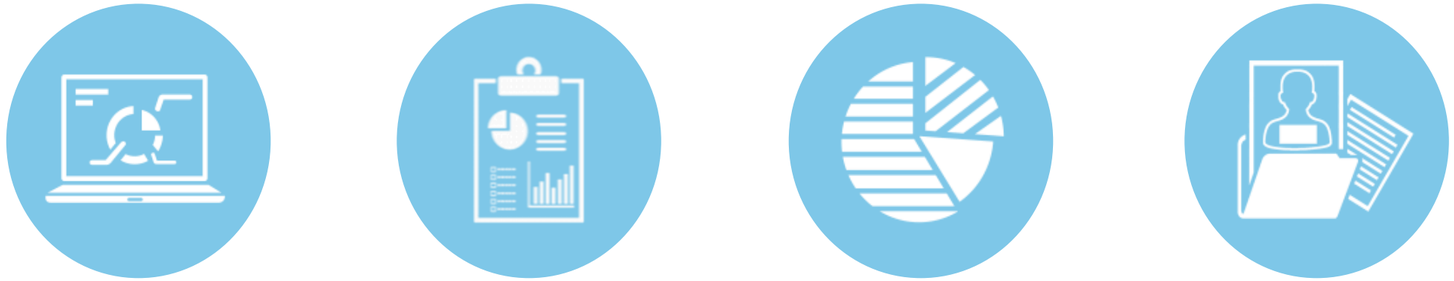
DevOps Certification Training

Lesson 00: Course Introduction



© Simplilearn. All rights reserved.

Course Objectives

By the end of this course, you will be able to acquire the following skills:

- Describe DevOps & DevSecOps methodologies and their key concepts.
- Explain the types of version control systems, continuous integration tools, continuous monitoring tools, and cloud models.
- Work with: Git with GitHub and Git with Bitbucket.
- Configure your private Jenkins, TeamCity tools, and Jenkins with Java, Git, and Maven.
- Set up a test-driven development framework with Junit 5 and a behavior-driven development framework with cucumber.
- Describe the importance of Chef and Puppet. Perform demonstration on Ansible.
- Work on the creation of Docker containers, Registry, Docker Compose, Docker Hub, and Docker Networking.
- Describe the importance of Grafana and ELK Stack. Perform demonstration on Nagios.
- Describe the importance of cloud in DevOps, use of AWS in DevOps, and deploy your private Kubernetes cluster.
- Set up your complete private infrastructure using version control systems and CI/CD tools.

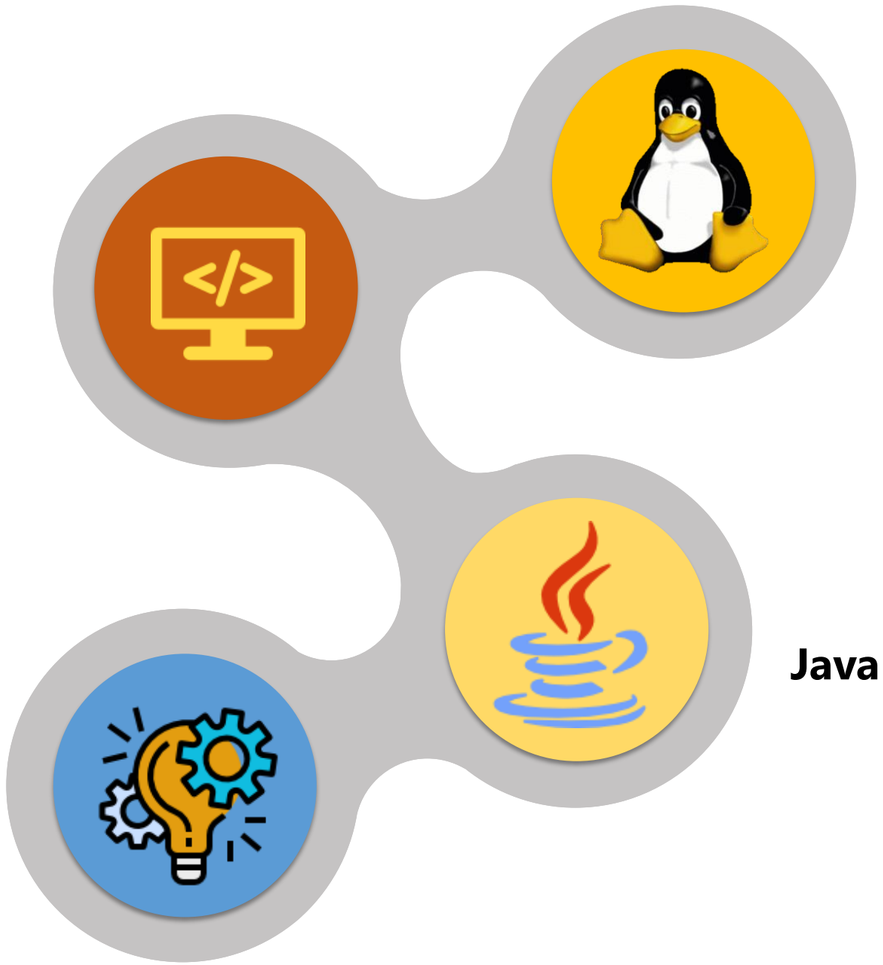
©Simplilearn. All rights reserved.

Course Outline

1. Introduction to DevOps
2. Version Control Systems
3. CI, CD, and Build Tools
4. Software and Automation Testing Frameworks
5. Configuration Management Tools
6. Containerization with Docker
7. Continuous Monitoring Tools
8. Need of Cloud in DevOps

©Simplilearn. All rights reserved.

Prerequisites for This Course



Linux Fundamentals

Web Development Fundamentals

Java Fundamentals

Project Management Fundamentals

©Simplilearn. All rights reserved.

Highlights of This Course

Agile Examples, DevOps, and DevSecOps Concepts

Set Up Git and Commit Files to GitHub and Bitbucket

Configure Jenkins, TeamCity, Docker Containers, Docker Compose, and Docker Hub

Getting Familiar with TDD with JUnit5 and BDD with cucumber

Set Up Kubernetes Cluster and Add Pods, Nodes, and Services

Multiple Hands-on Practices for Tools across Different Categories

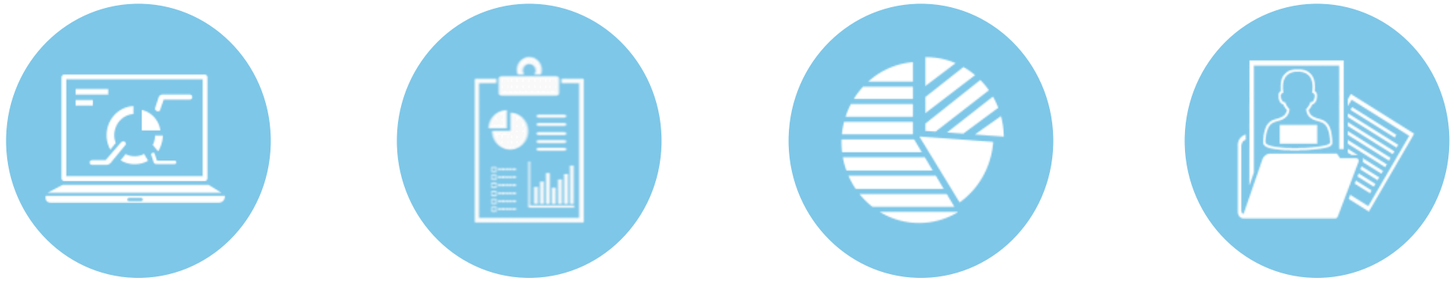
©Simplilearn. All rights reserved.

Thank You

©Simplilearn. All rights reserved.

DevOps Certification Training

Lesson 01: Introduction to DevOps



© Simplilearn. All rights reserved.

Learning Objectives

By the end of this lesson, you will be able to:

- State the principles of DevOps
- Explain the challenges in the traditional approach
- Describe how DevOps helps in overcoming challenges faced in traditional approach
- Describe the DevOps tools
- Explain DevSecOps and it's importance

©Simplilearn. All rights reserved.

Introduction to DevOps

DevOps Overview and Principles

©Simplilearn. All rights reserved.

DevOps

DevOps is short for **Dev**elopment and **Op**eration**s**. It concentrates on collaboration between developers and other parties involved in building, deploying, operating, and maintaining software systems.

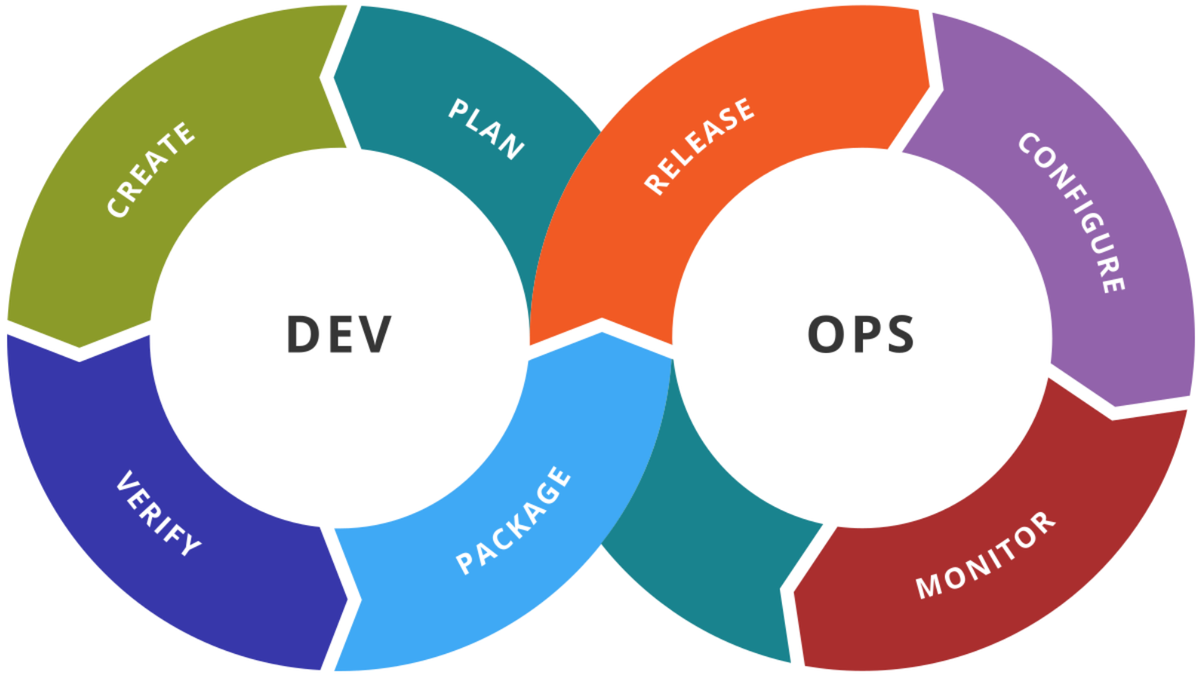
©Simplilearn. All rights reserved.

History of DevOps

- Patrick Debois, a Belgian consultant, project manager, and agile practitioner is one among the initiators of DevOps.
- A presentation on “10+ Deploys per Day: Dev and Ops Cooperation at Flickr” helped in bring out the ideas for DevOps and resolve the conflict of “ It’s not my code, it’s your machines! ”
- DevOps blends lean thinking with agile philosophy.

©Simplilearn. All rights reserved.

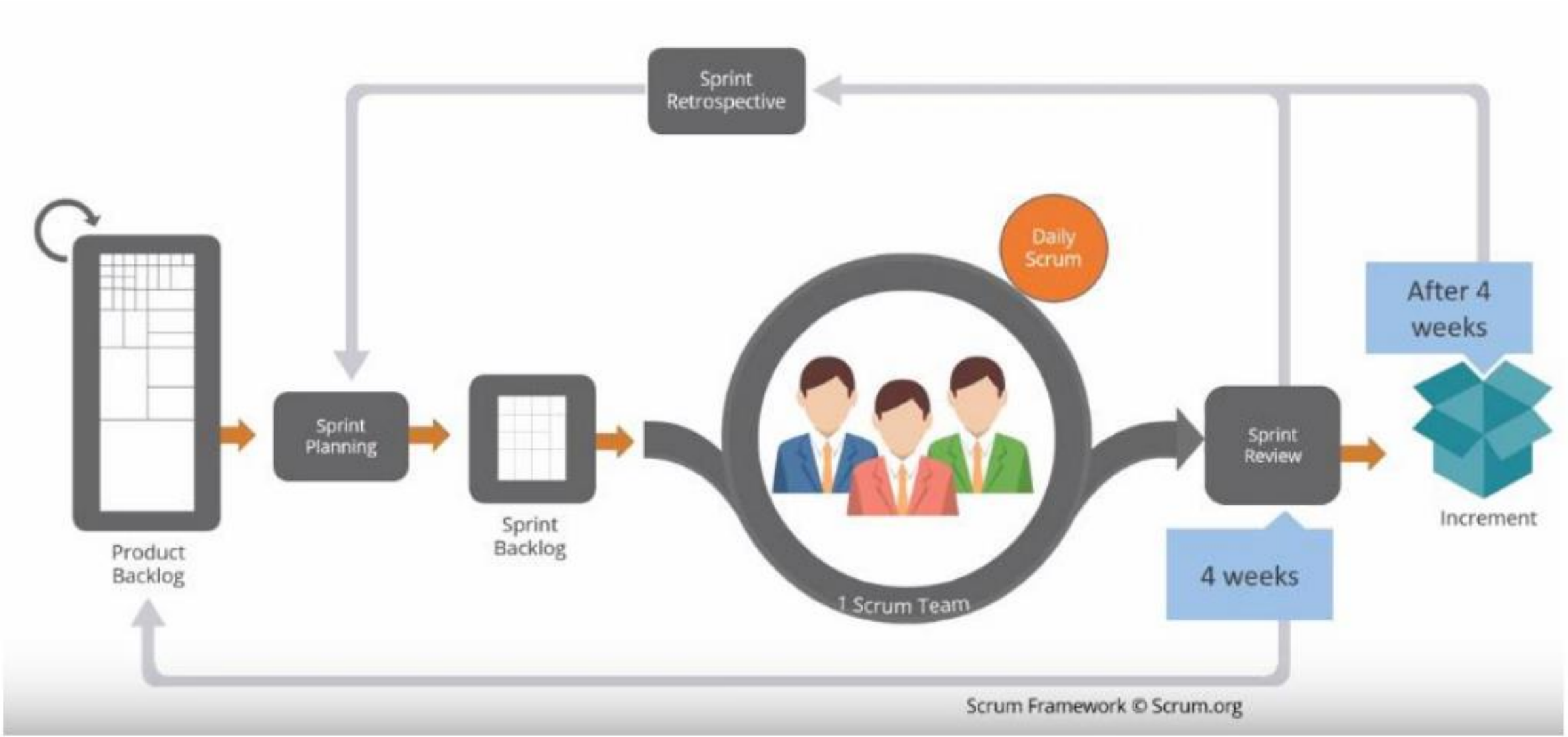
Overview of DevOps



- DevOps is an agile relationship between development and IT operations
- DevOps is the abbreviation for **Dev**elopment and **Op**erations
- The Development includes Plan, Create, Verify, and Package
- The Operations include Release, Configure, and Monitor

©Simplilearn. All rights reserved.

Agile Example: Scrum



©Simplilearn. All rights reserved.

Relationship Between Agile and DevOps

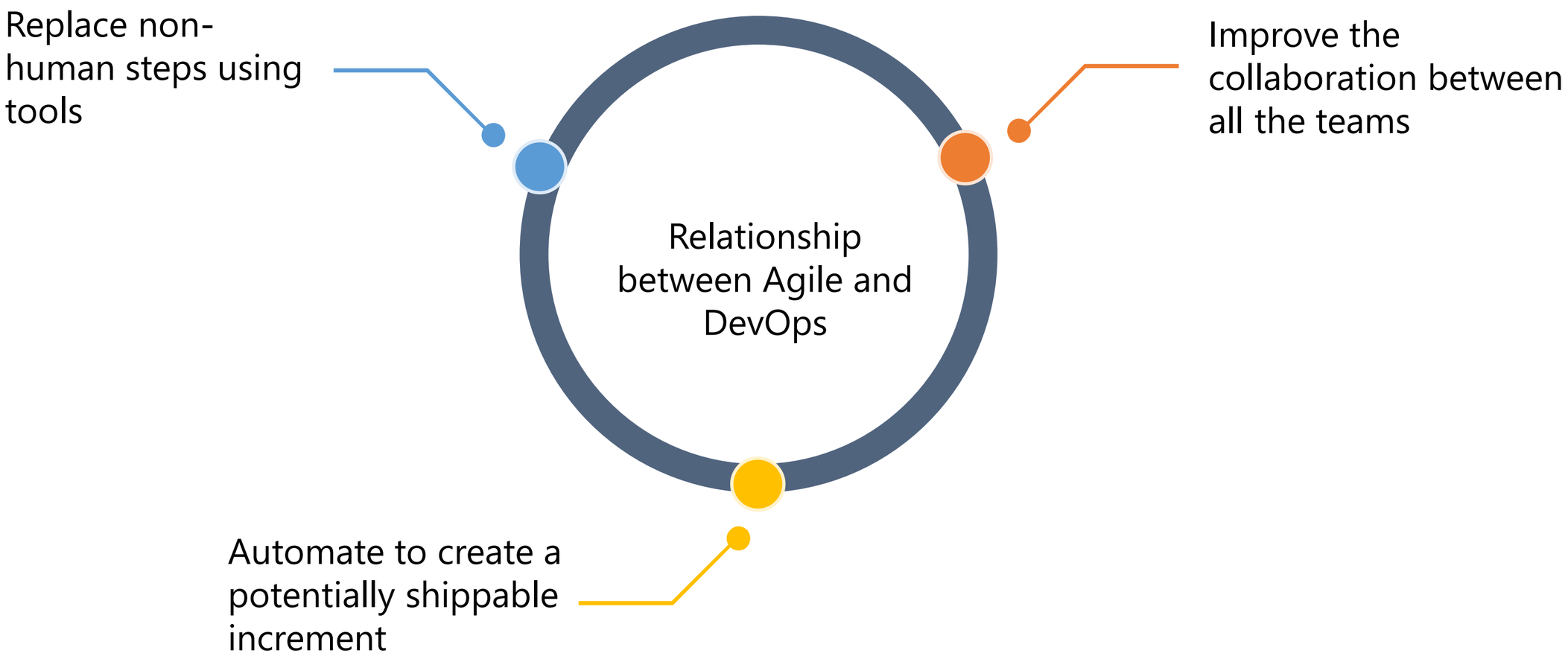
Satisfy the customer through early and continuous delivery of valuable software

Deliver working software frequently with a preference for the shorter timescale

Business people and developers must work together daily throughout the project

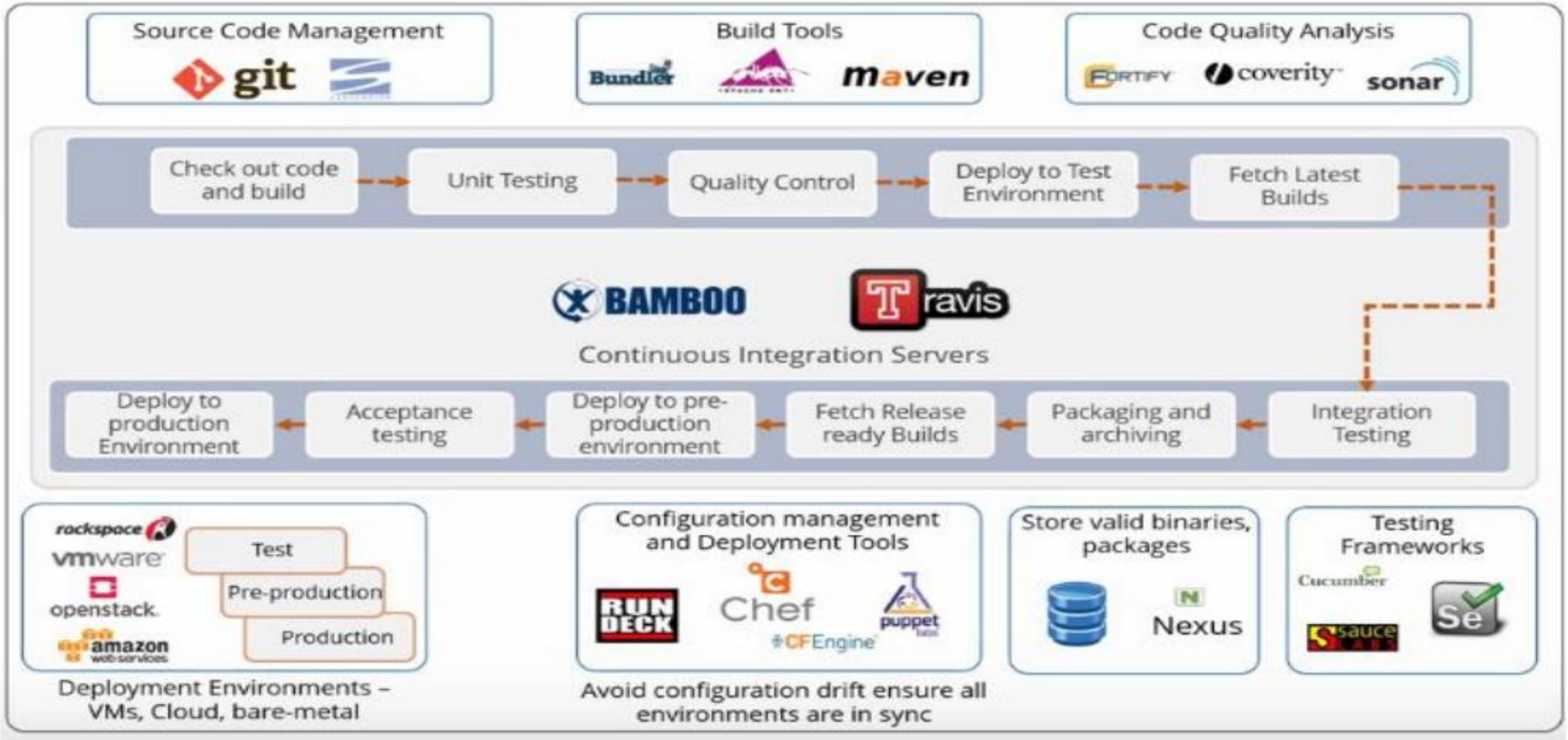
©Simplilearn. All rights reserved.

Relationship Between Agile and DevOps



©Simplilearn. All rights reserved.

Agile and DevOps Example



©Simplilearn. All rights reserved.

DevOps Toolchains

Monitoring Performance

Releasing into Production

Building Applications

Code Development and Unit Testing

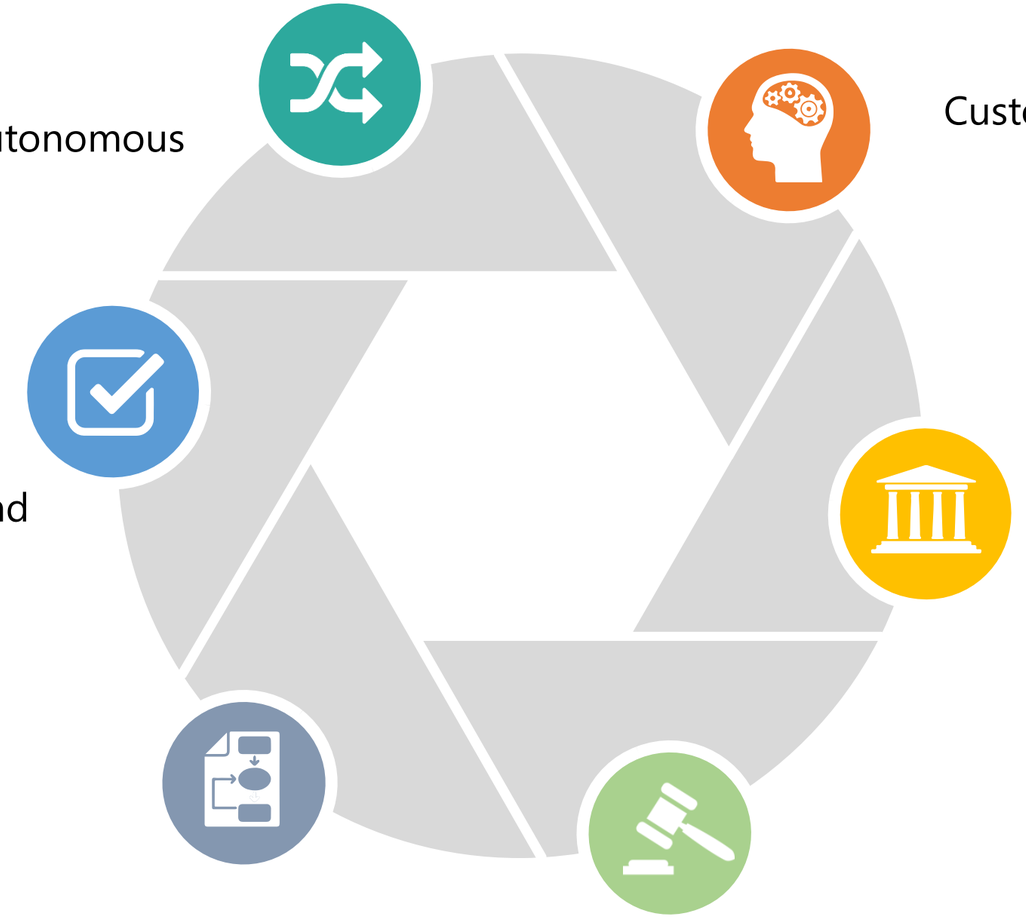
Configuration Management

Integration and Performance Testing

Packing the Application

©Simplilearn. All rights reserved.

DASA DevOps Principles



Cross-functional autonomous teams

Customer-centric action

Create with the end in mind

End-to-end responsibility

Continuous improvement

Automate everything you can

©Simplilearn. All rights reserved.

Case Study

Amazon switched to DevOps at the development phase of their web services, popularly known as AWS.

©Simplilearn. All rights reserved.

Introduction to DevOps

Challenges in the Traditional Approach

©Simplilearn. All rights reserved.

Waterfall Method

Most of the development teams use waterfall method, which is time-consuming because of the larger size of the developer team, testers, and the code involved.

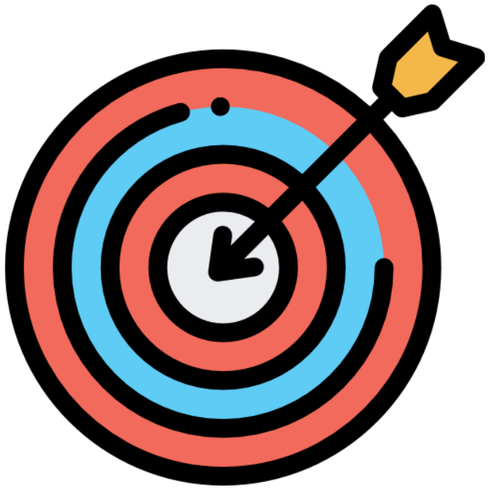
©Simplilearn. All rights reserved.

Productivity

Codes that are large and bundled into release will result in jammed production and lower the productivity.

©Simplilearn. All rights reserved.

Difficult to Achieve Goal



Less investment on resources and constant work make it difficult for the developers to achieve goal or an outcome.

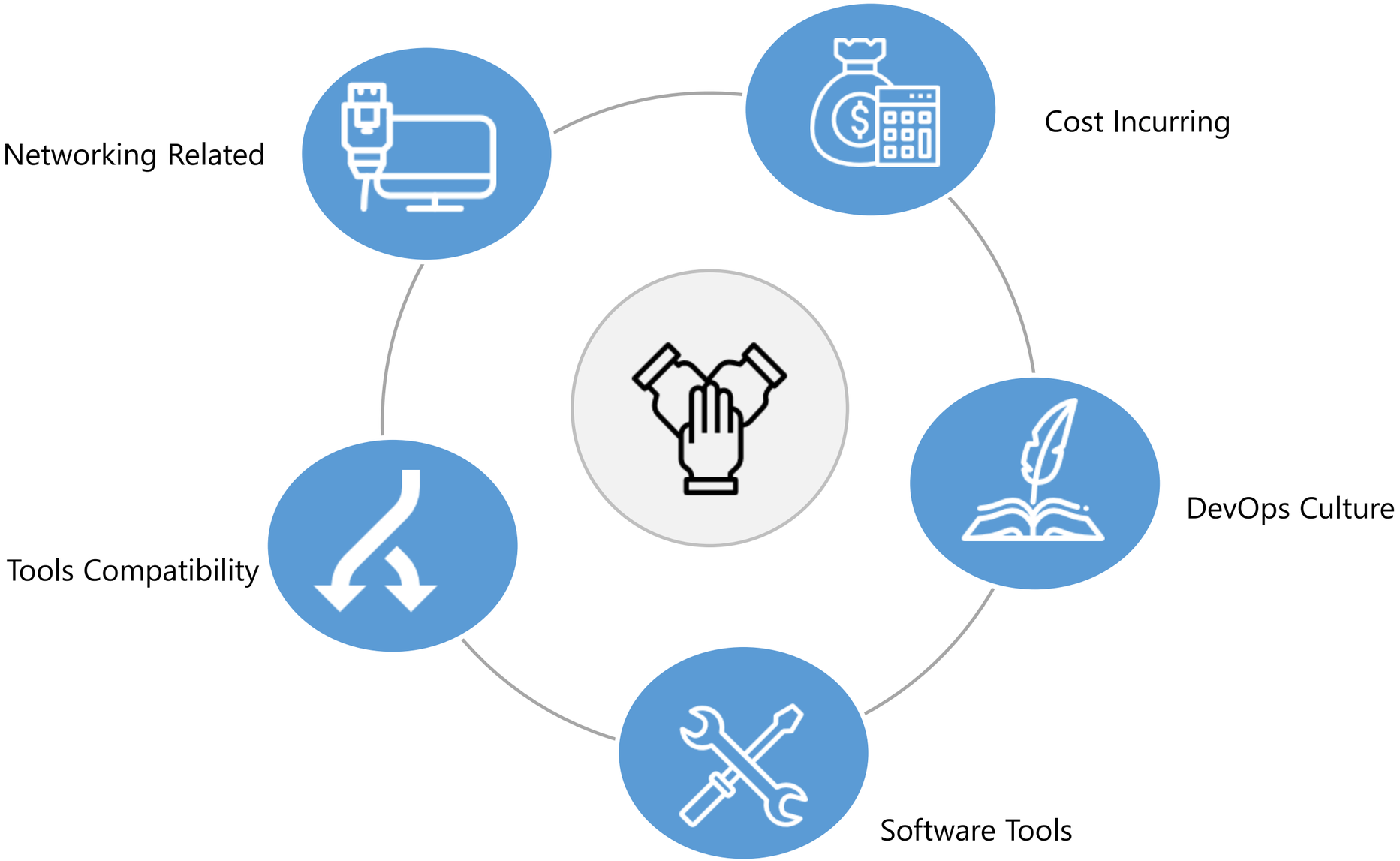
©Simplilearn. All rights reserved.

Investment in Schedule Planning Systems

More money is invested in schedule planning systems which are sensitive and inaccurate. As a result, it consumes more time to manage the systems.

©Simplilearn. All rights reserved.

Constantly Changing Challenges



©Simplilearn. All rights reserved.

Types of Constantly Changing Challenges

DevOps Culture	Adapts to continuous changes : People's resistance, organization culture and pair-programming
Software Tools	Preference on tools across teams, standardization of tools across the organization in terms of licensing, version incompatibilities, maintenance and support
Tools Compatibility	Existence of legacy systems. Hybrid environments and tools must be cloud ready
Network Related Issues	Internet Connectivity, Data Center Connectivity, Virtual LANs, Cloud Connectivity
Cost Related Issues	New hardware requirements, software licensing, training, reduced efficiency while learning

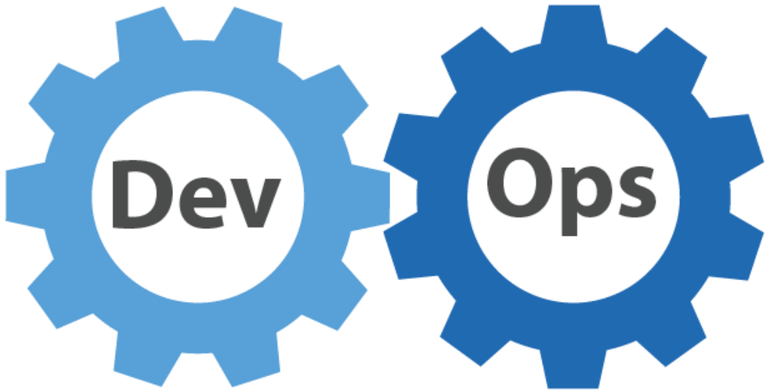
©Simplilearn. All rights reserved.

Introduction to DevOps

DevOps Approach to the Challenges

©Simplilearn. All rights reserved.

DevOps Way to Overcome Challenges



- Reduction in the code size delivery results in increased productivity
- Smaller batch sizes, dedicated teams, and automated processes make scheduling simpler to operate
- Batch sizes are divided into small cells. Each cell gathers its own data, reducing the size of the reports
- Identifies productive and loss areas in the process. As a result, an organization can focus more on their goals

©Simplilearn. All rights reserved.

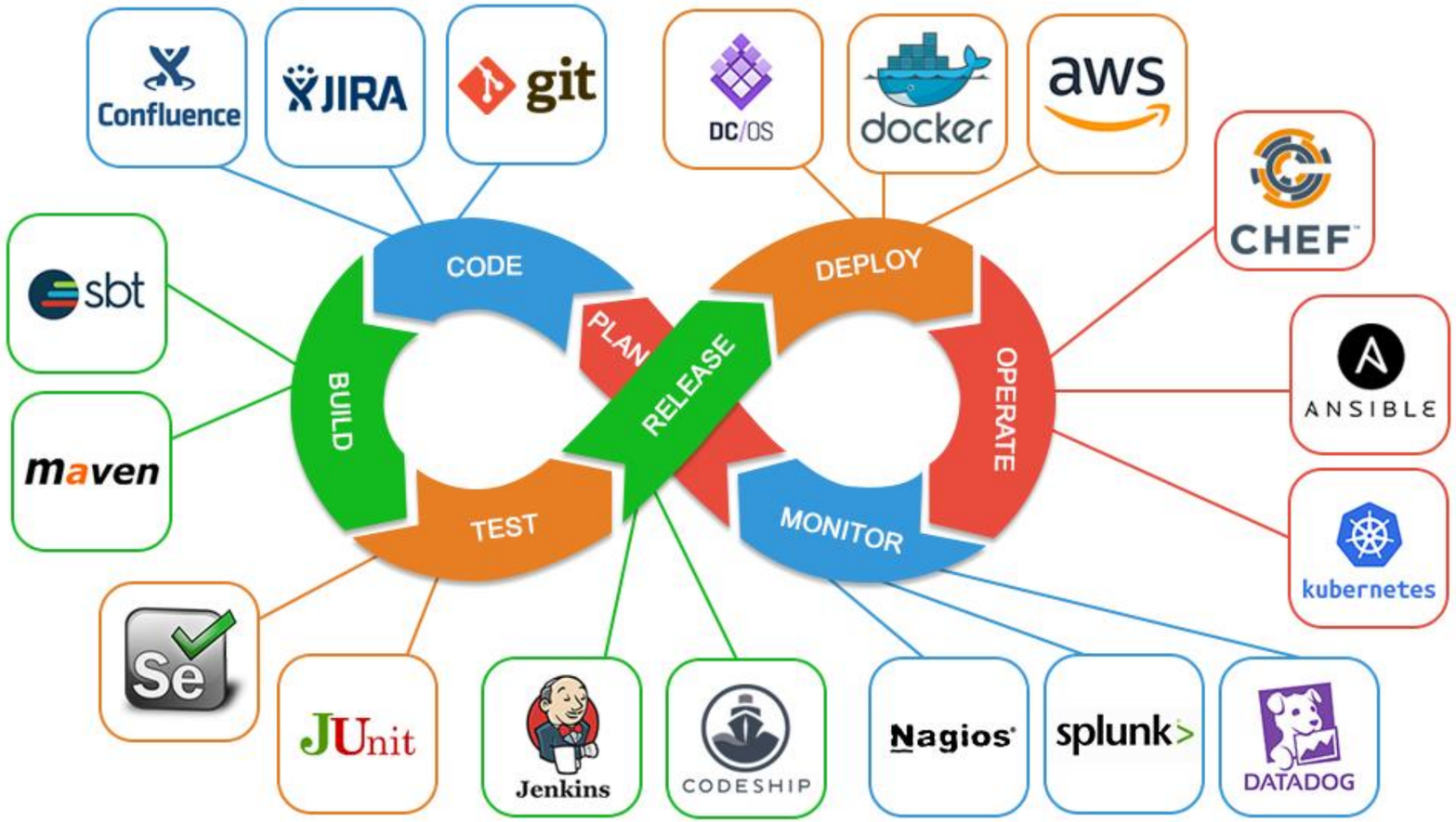
Introduction to DevOps

Overview of DevOps Tools

©Simplilearn. All rights reserved.

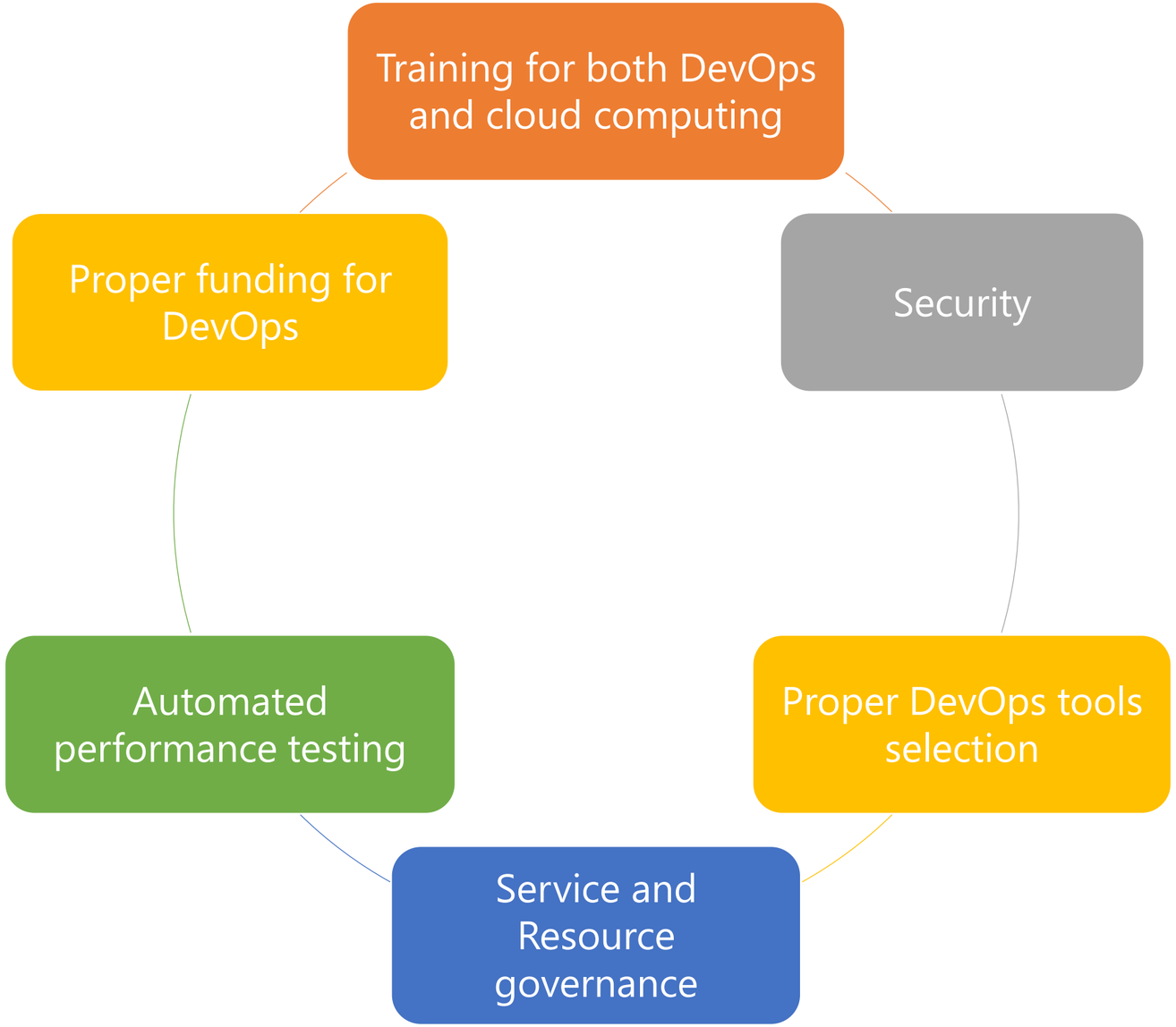
DevOps Tools

To implement DevOps and work within the DevOps setup, the various tools required are:



©Simplilearn. All rights reserved.

Best Practices for DevOps



©Simplilearn. All rights reserved.

DevOps and Cloud Computing Training

Proper training in DevOps and cloud computing will lead to better understanding and help resolve challenges in DevOps environment.

©Simplilearn. All rights reserved.

Security

Security should be part of the automated testing. It should be built into continuous integration and deployment processes during the migration to cloud-based platform.

©Simplilearn. All rights reserved.

Proper DevOps Tools Selection

The Applications should be deployable on different clouds. In this way, you can pick and choose the best public or private cloud for the job.

Open source and licensed tools

Tools that work together

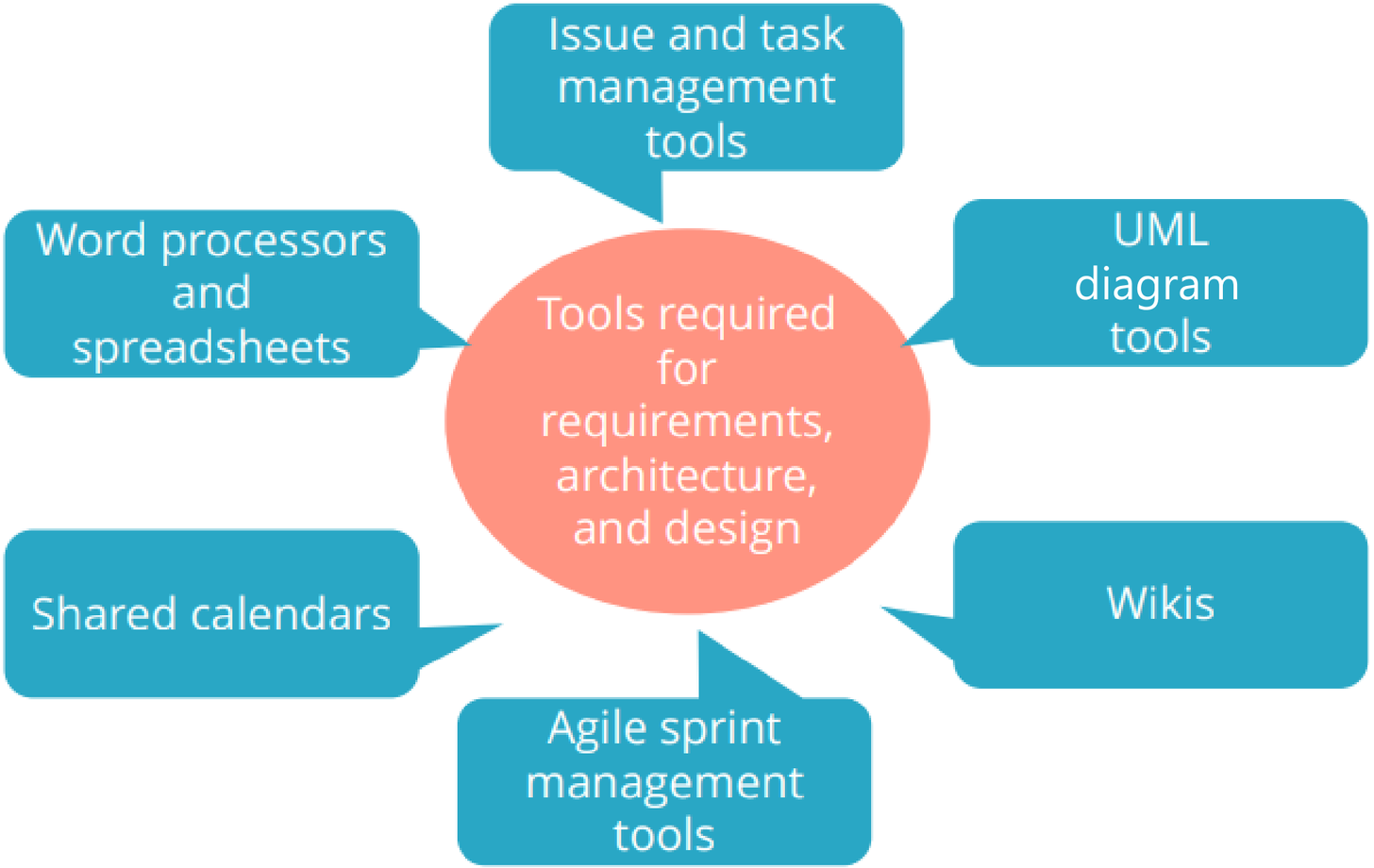
Tools for different SDLC phases

Tools that perform similar functions

©Simplilearn. All rights reserved.

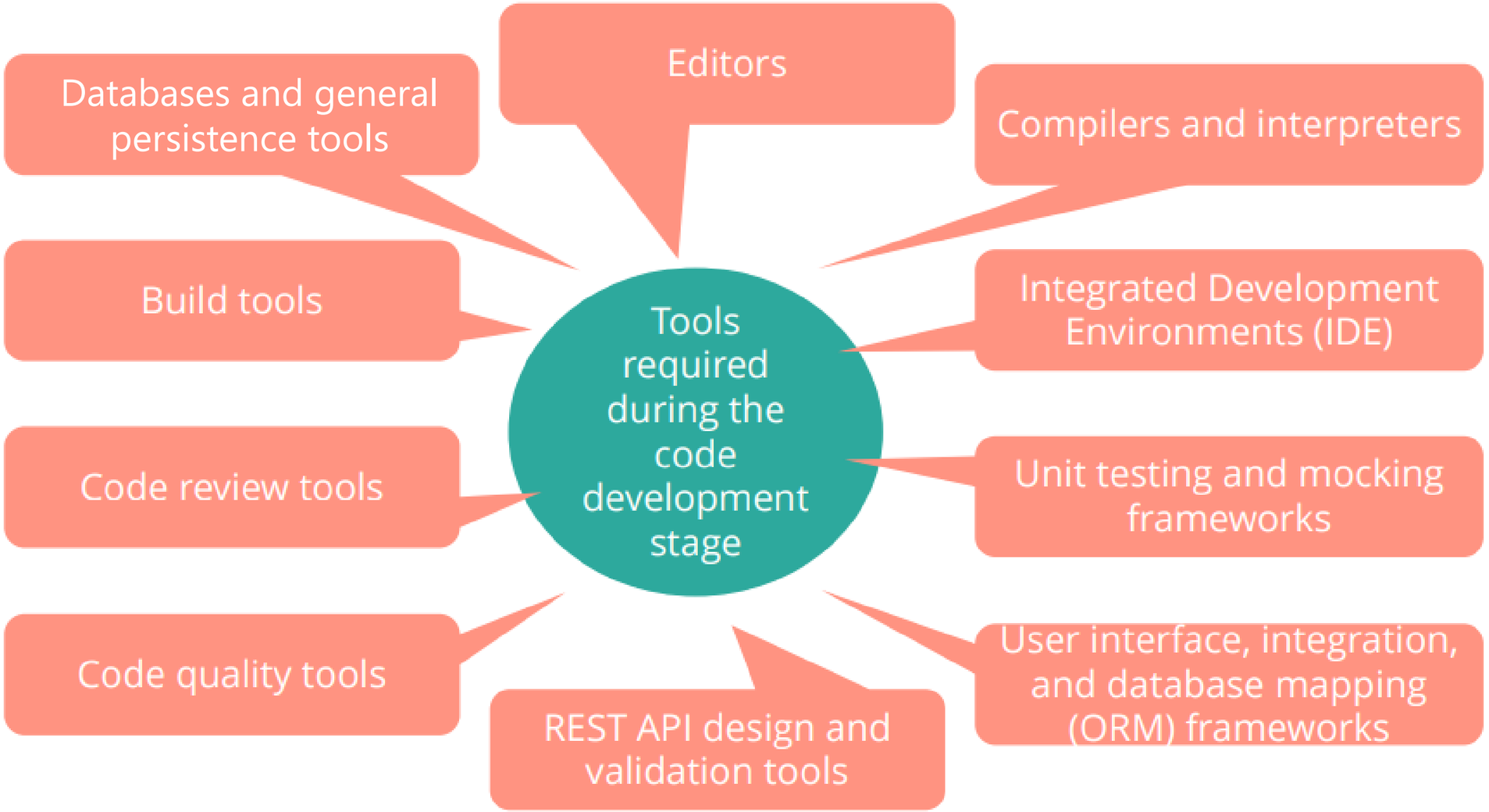
Requirements Tools

Tools are used to share the files and communicate within the team and other teams.



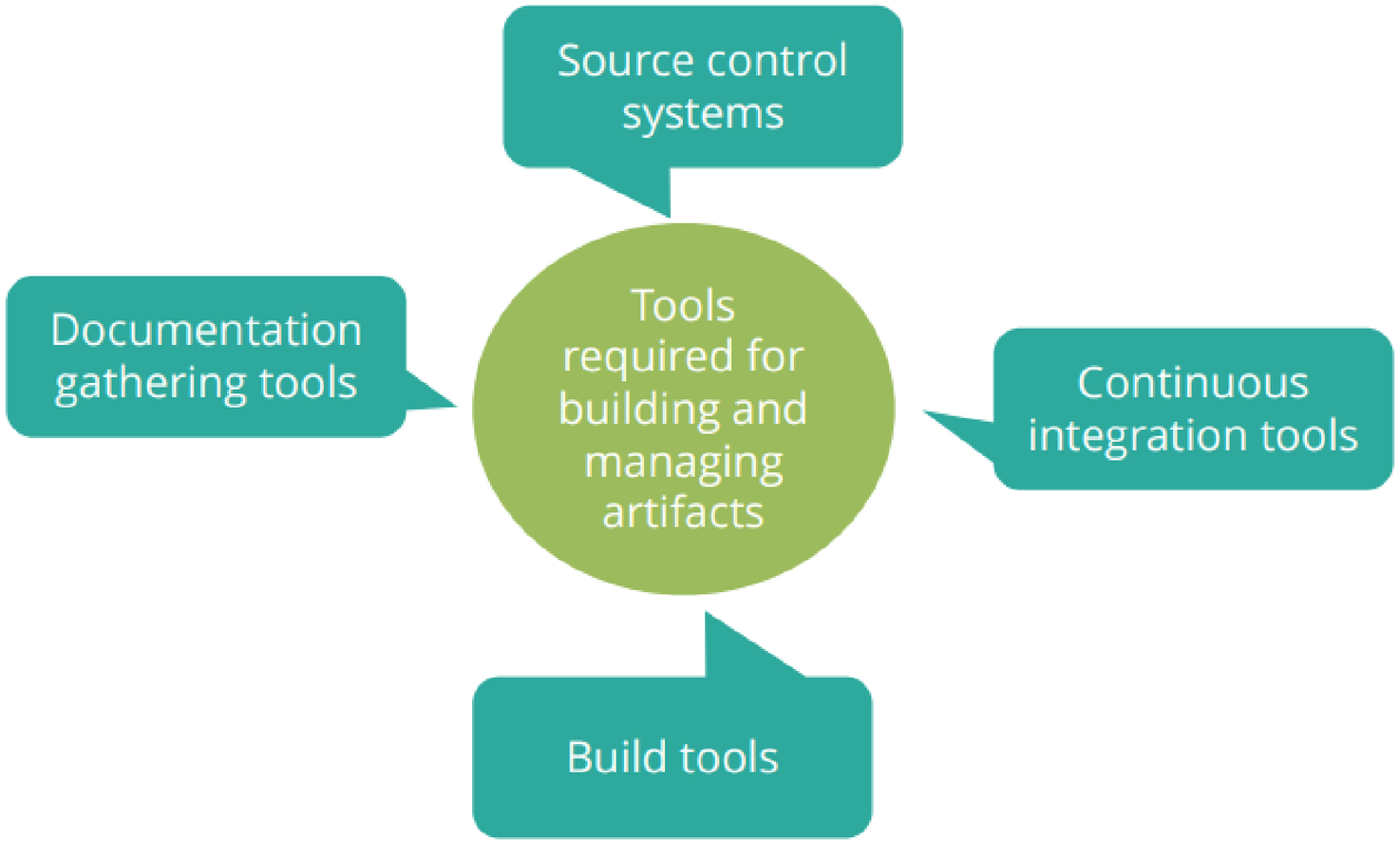
©Simplilearn. All rights reserved.

Code Development Tools



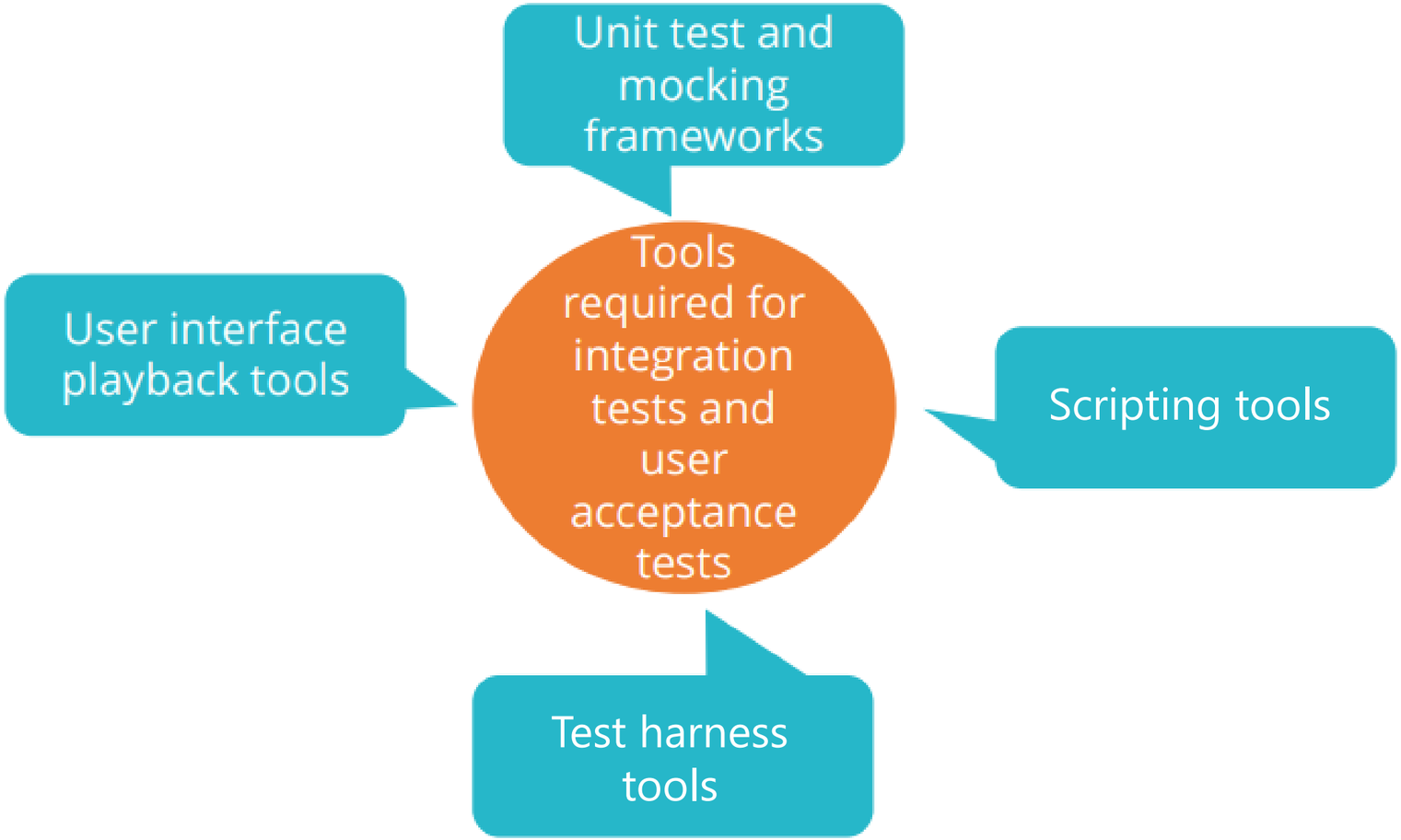
©Simplilearn. All rights reserved.

Artifact Creation Tools



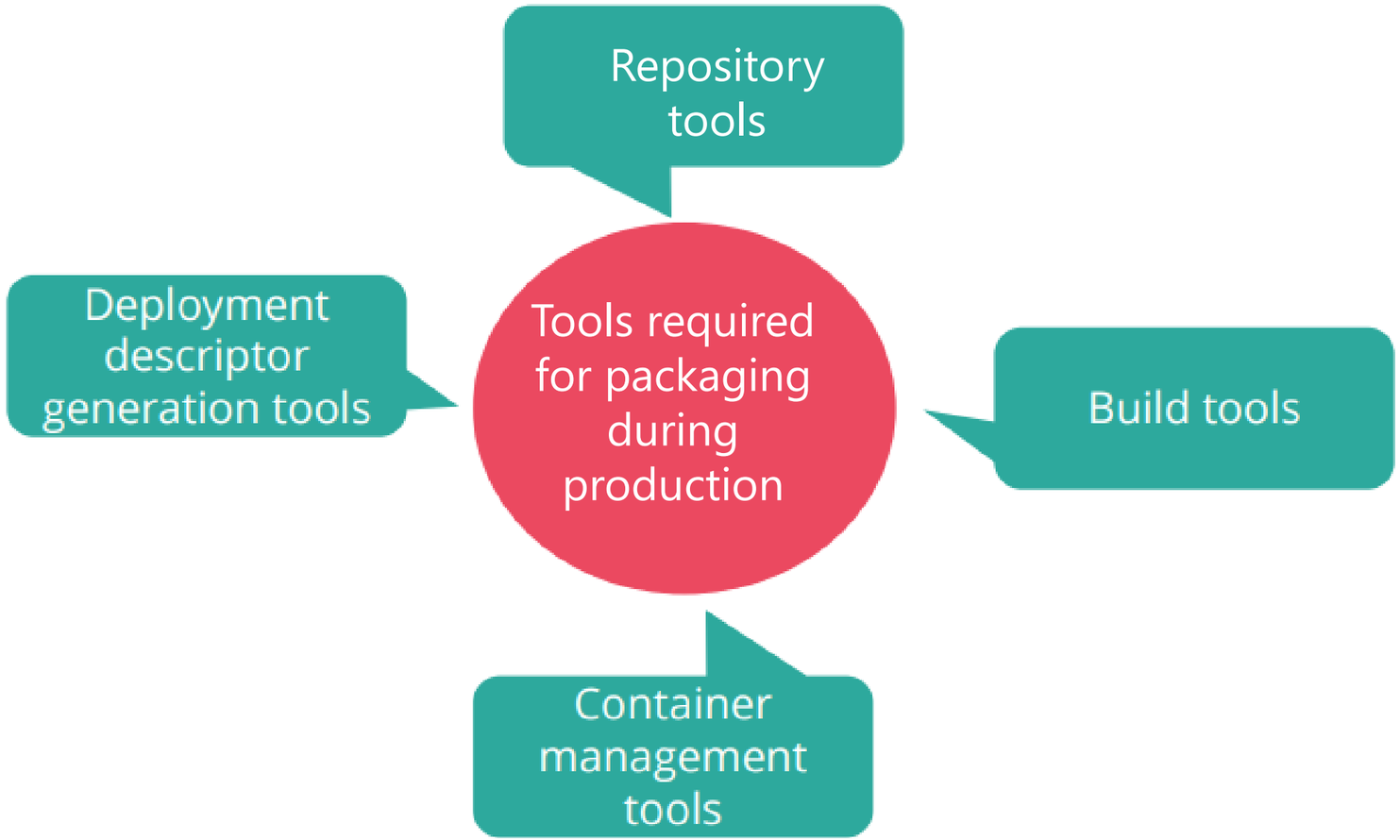
©Simplilearn. All rights reserved.

Testing Tools



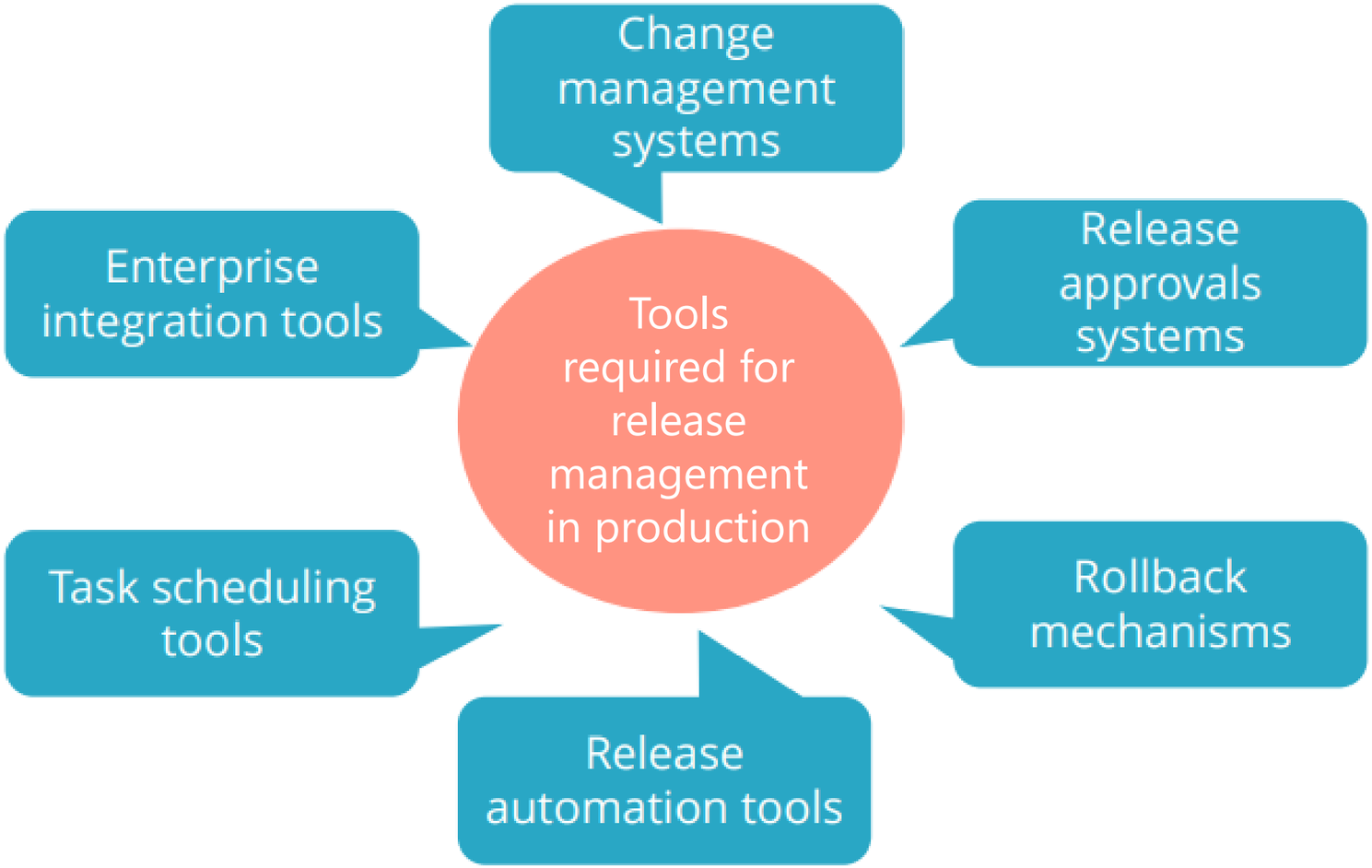
©Simplilearn. All rights reserved.

Packaging Tools



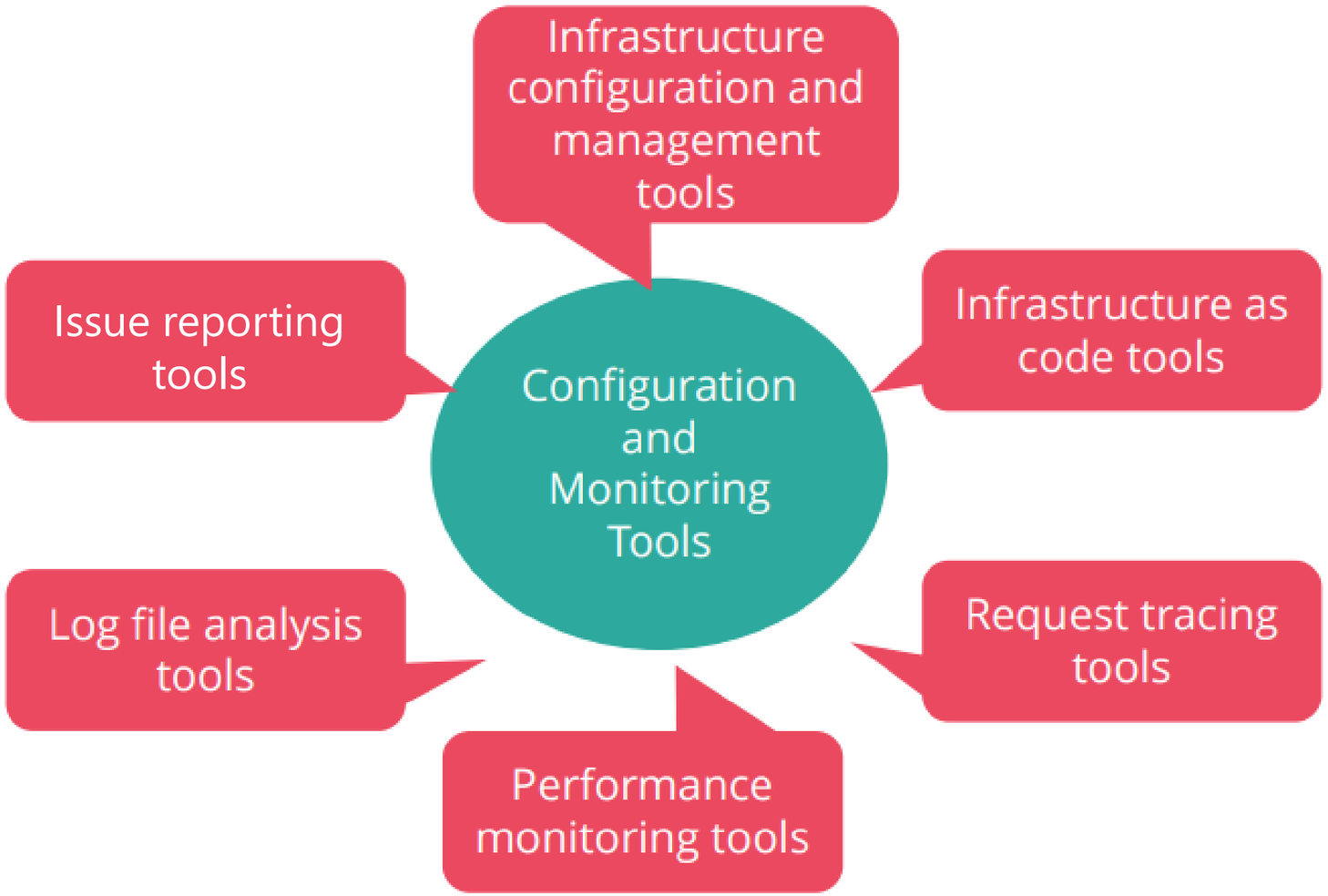
©Simplilearn. All rights reserved.

Release Management Tools



©Simplilearn. All rights reserved.

Configuration and Monitoring Tools



©Simplilearn. All rights reserved.

Introduction to DevOps

DevSecOps and It's importance

©Simplilearn. All rights reserved.

DevSecOps

DevSecOps is the ideology of integrating security practices within the DevOps culture.

01 Creates a ‘Security as Code’ Culture

02 Creates a flexible collaboration between the team members

03 Bridges traditional gaps between IT and Security

04 Aims to embed security in every part of the development process

©Simplilearn. All rights reserved.

DevSecOps Manifesto

Leaning in over Always Saying “no”

Data & Security Science over Fear, Uncertainty and Doubt

Open Contribution & Collaboration over Security-Only Requirements

Consumable Security Services with APIs over Mandated Security Controls & Paperwork

Business Driven Security Scores over Rubber Stamp Security

Red & Blue Team Exploit Testing over Relying on Scans & Theoretical Vulnerabilities

24*7 Proactive Security Monitoring over Reacting after being informed of an incident

Shared Threat Intelligence over Keeping Info to Ourselves

Compliance Operations over Clipboards & Checklists

©Simplilearn. All rights reserved.

Workflow of DevOps and DevSecOps

- A Developer writes a code for a feature within the version control management system
- The required changes are committed to the version control management system
- Another Developer retrieves the code from the version control management system and performs analysis of the code to identify security threats
- An environment is created using any management tool such as puppet or chef. The application is deployed and security configurations are applied to the system.

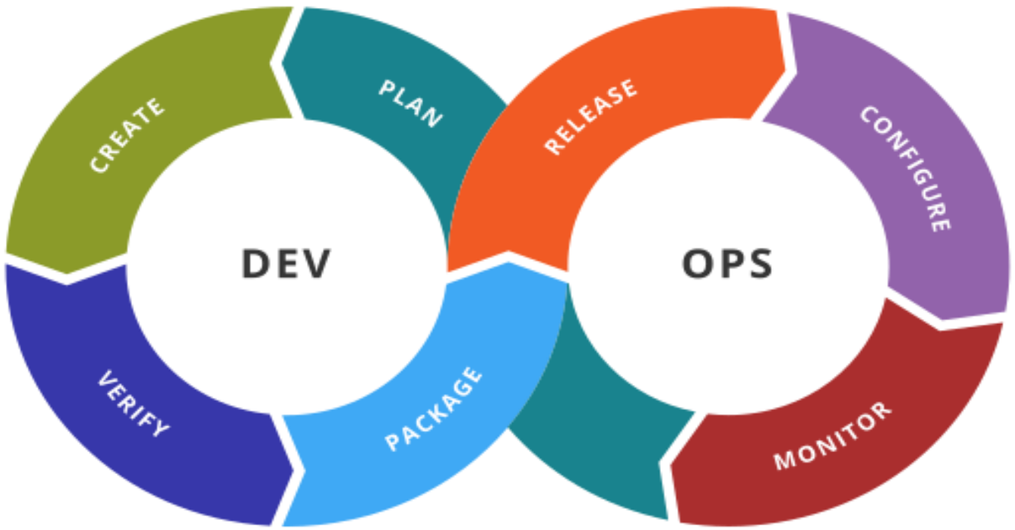
©Simplilearn. All rights reserved.

Workflow of DevOps and DevSecOps (Contd.)

- A test automation suite is executed against the deployed application on factors like UI, integration, security tests, and APIs
- If the application passes the test, it is deployed to the production environment
- The newly production environment is monitored continuously to identify any active security threats

©Simplilearn. All rights reserved.

DevOps vs. DevSecOps



Features:

Speed

Rapid delivery

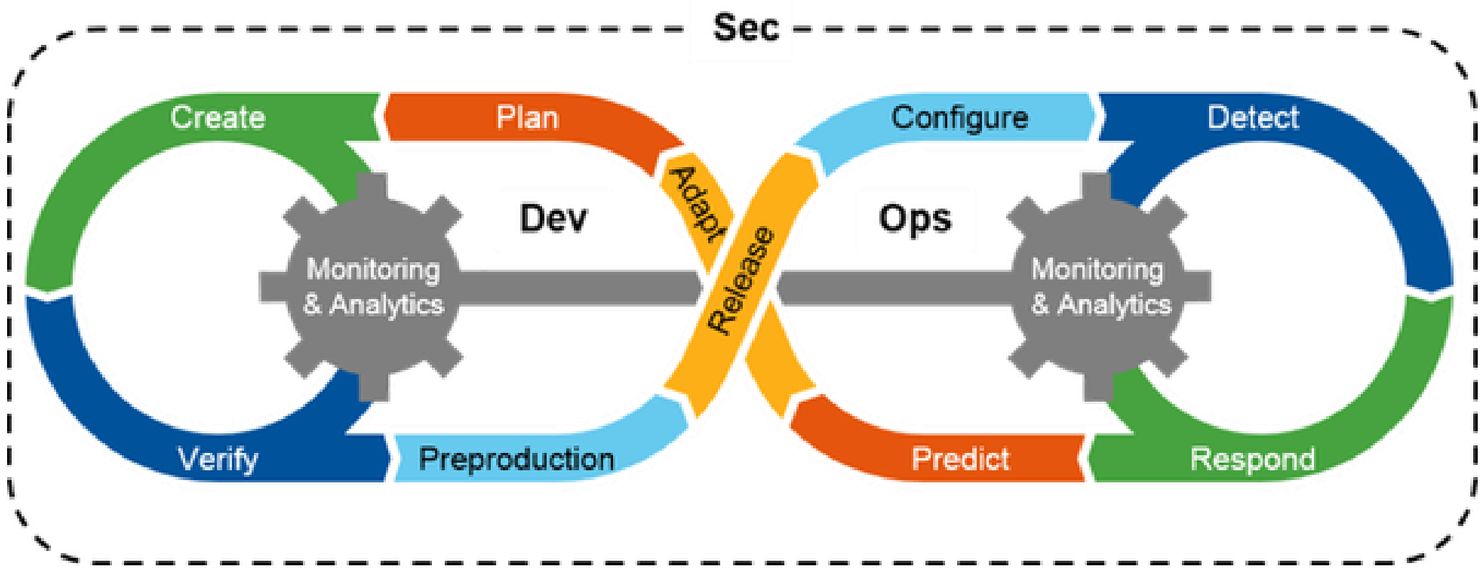
Scaled and less secured

Features:

Agility

Security automation

Security as code



©Simplilearn. All rights reserved.

Key Takeaways

You are now able to:

- State the principles of DevOps
- Explain the challenges in the traditional approach
- Describe how DevOps helps in overcoming challenges faced in traditional approach
- Describe the DevOps tools
- Explain DevSecOps and it's importance

©Simplilearn. All rights reserved.

Knowledge Check

©Simplilearn. All rights reserved.

Knowledge Check 1	**Which one of the following methodologies has least impact on DevOps methodology?**

a. Lean Manufacturing

b. Agile Software Delivery

c. Waterfall Software Delivery

d. Continuous Software Delivery

©Simplilearn. All rights reserved.

Knowledge Check 1	**Which one of the following methodologies has least impact on DevOps methodology?**

a. Lean Manufacturing

b. Agile Software Delivery

c. Waterfall Software Delivery

d. Continuous Software Delivery

The correct answer is **c. Waterfall Software Delivery**

©Simplilearn. All rights reserved.

Knowledge Check 2	**Which one of the following techniques makes DevOps a successful methodology to develop and deliver software?**

a. DevOps helps in organizing the teams and organizational mission

b. DevOps helps in creating the software with built-in quality and monitoring

c. DevOps helps to quickly identify, fix and learn from errors

d. All of the above

©Simplilearn. All rights reserved.

Knowledge Check 2	**Which one of the following techniques makes DevOps a successful methodology to develop and deliver software?**

a. DevOps helps in organizing the teams and organizational mission

b. DevOps helps in creating the software with built-in quality and monitoring

c. DevOps helps to quickly identify, fix and learn from errors

d. All of the above

The correct answer is **d. All of the above**

©Simplilearn. All rights reserved.

Knowledge Check 3	**Which is the best technique to convert normal changes into standard changes?**

a. Use the existing track record of successful automated deployments with standard changes

b. Negotiate with release managers

c. Publicly complain about bureaucracy and make everyone be aware of it

d. Make sure normal changes are very carefully deployed to the production

©Simplilearn. All rights reserved.

Knowledge Check 3	**Which is the best technique to convert normal changes into standard changes?**

a. Use the existing track record of successful automated deployments with standard changes

b. Negotiate with release managers

c. Publicly complain about bureaucracy and make everyone be aware of it

d. Make sure normal changes are very carefully deployed to the production

The correct answer is **a. Use the existing track record of successful automated deployments with standard changes**

©Simplilearn. All rights reserved.

Knowledge Check 4	**Which is the widely reusable asset to reinforce information security?**

a. Data Storage Systems

b. Tools handling the logging of sensitive client information

c. Transfer of data between clients and software

d. All of the above

©Simplilearn. All rights reserved.

Knowledge Check 4	**Which is the widely reusable asset to reinforce information security?**

a. Data Storage Systems

b. Tools handling the logging of sensitive client information

c. Transfer of data between clients and software

d. All of the above

The correct answer is **d. All of the above**

©Simplilearn. All rights reserved.

Knowledge Check 5	**What is the benefit of using feature toggles embedded in configuration of the application during fixing errors?**

a. Easiest way to fix a problem

b. Don’t have to correct erroneous pieces immediately during deployment

c. DevOps team can take time to properly identify root cause of an issue and improve the techniques

d. All of the above

©Simplilearn. All rights reserved.

Knowledge Check 5	**What is the benefit of using feature toggles embedded in configuration of the application during fixing errors?**

a. Easiest way to fix a problem

b. Don’t have to correct erroneous pieces immediately during deployment

c. DevOps team can take time to properly identify root cause of an issue and improve the techniques

d. All of the above

The correct answer is **d. All of the above**

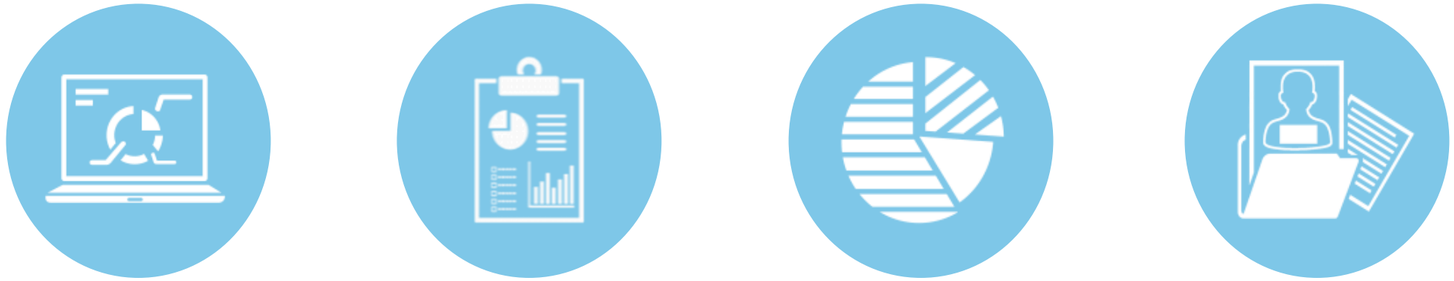
©Simplilearn. All rights reserved.

Thank You

©Simplilearn. All rights reserved.

DevOps Certification Training

Lesson 02: Version Control Systems



© Simplilearn. All rights reserved.

Learning Objectives

By the end of this lesson, you will be able to:

- Select the suitable version control systems for your organization
- Explain the role of version control systems
- Identify the types of control systems and the supporting tools
- Set up Git
- Identify the suitable source code and version control hosts

©Simplilearn. All rights reserved.

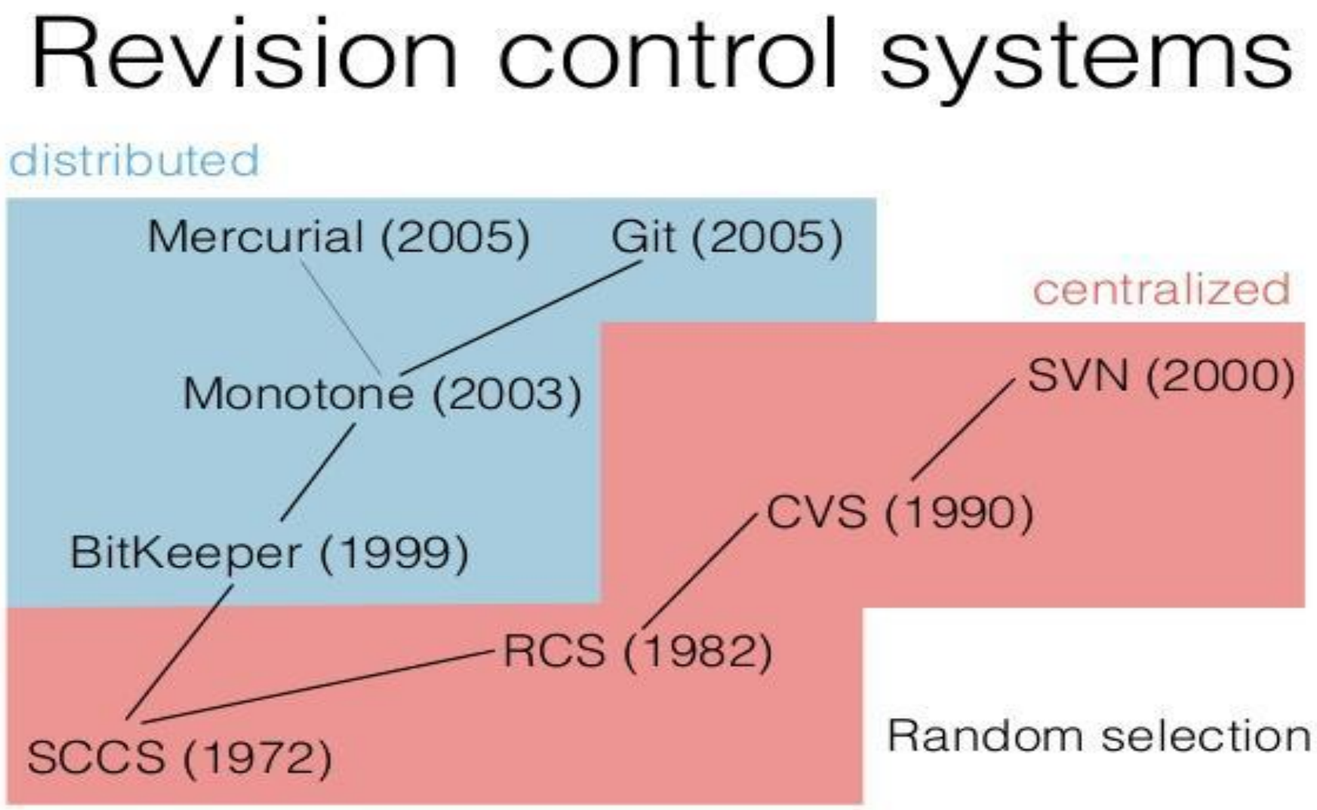
Version Control Systems

Overview of Version Control Systems

©Simplilearn. All rights reserved.

History of Version Control Systems

A major focus of version control system (also known as revision control or source control) is to manage the changes to the files, programs, logs, and other information related to code development, code deployment, and code operation.



©Simplilearn. All rights reserved.

Need of Version Control System

Version control systems can be used to accomplish various tasks.

Files are checked into version control by registered users only

The username and timestamp of the check-in are associated with the revision

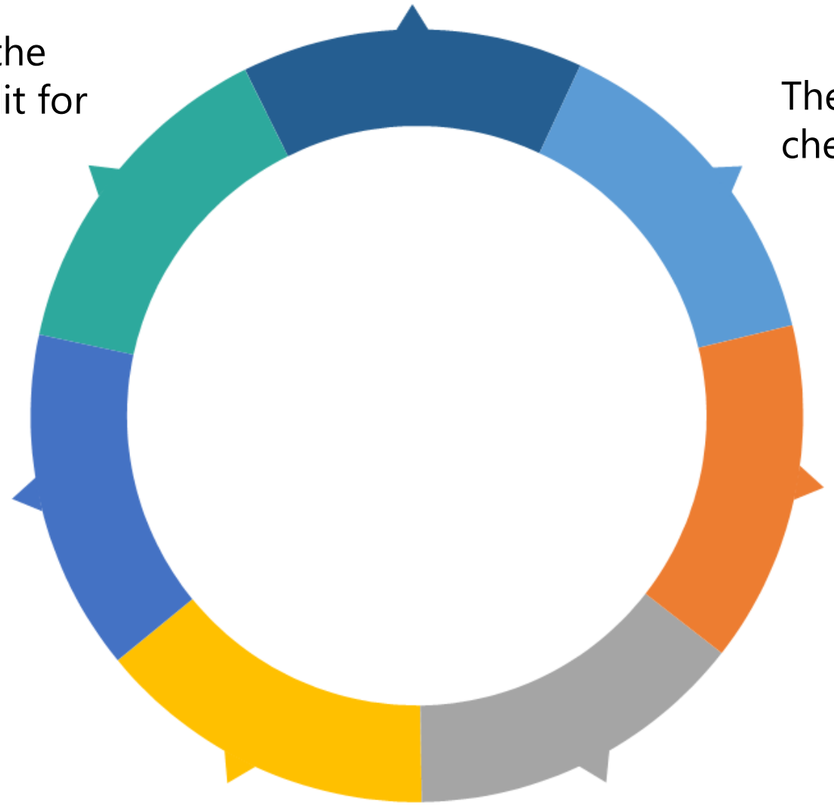
Each file check-in gets a new version which is usually a number

All the previous versions of a file can be easily extracted

The latest versions of all files are often referred to as heads

Source control systems are organized into repositories

Supports branching where the head of the repository is split for parallel development



©Simplilearn. All rights reserved.

Repository Usage

1

Files and folders which need to be changed over time.
Example: Constant addition of code for every increment in the feature.

2

Better tools for replicating data which doesn't change.
Example: Time machine for Apple computers.

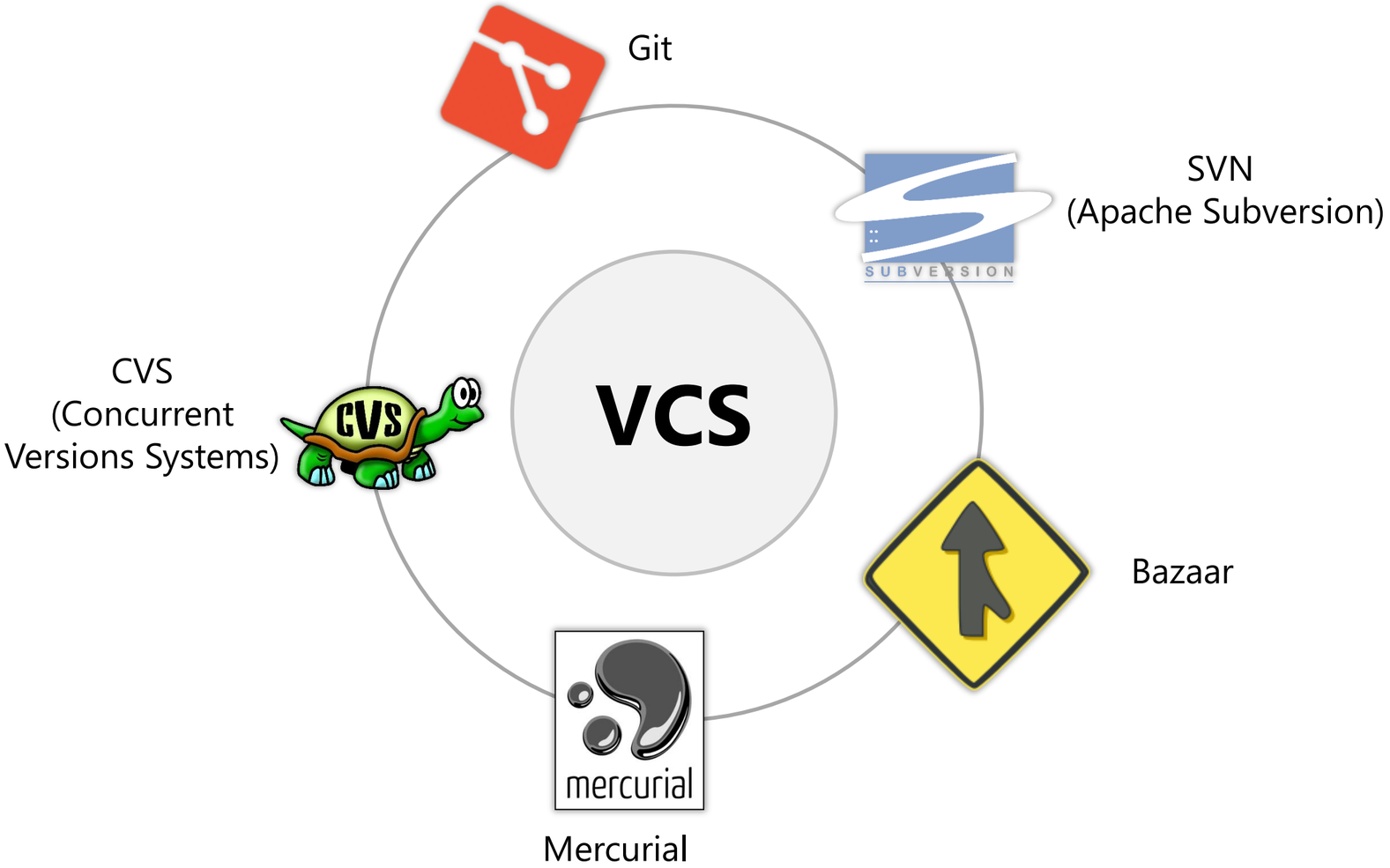
3

Inappropriate use of a repository. Example: A photograph or music library.

©Simplilearn. All rights reserved.

Popular Version Control Systems

Some of the most preferred and popular open-source version control systems and tools are listed:



©Simplilearn. All rights reserved.

Version Control Systems

Role of Version Control Systems

©Simplilearn. All rights reserved.

Importance of Version Control System

1. Helps in tracing the development of the documents such as policies or source codes
2. All the documents can be identified by filenames, authors, and the modified dates. It helps in linking contributions and contributors
3. Helps in differentiating the latest versions and track the recent updates
4. The “truth” files and documents can be stored in a given storage location
5. Helps in keeping track of the independent code versions for various reasons
6. Continuous integration is easier with the newer version control systems
7. Helps in coding collaboratively and remotely

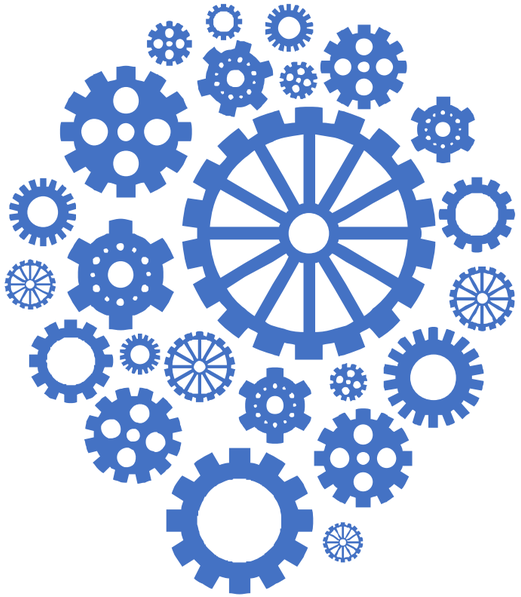
©Simplilearn. All rights reserved.

Role of Version Control System in DevOps Environment

Organize product-related artifacts

Adopt peer reviews to inspect changes

Provide tried and tested guaranteed results



Regulate monitoring of documents

Establish win-win relationship between Dev and Ops

©Simplilearn. All rights reserved.

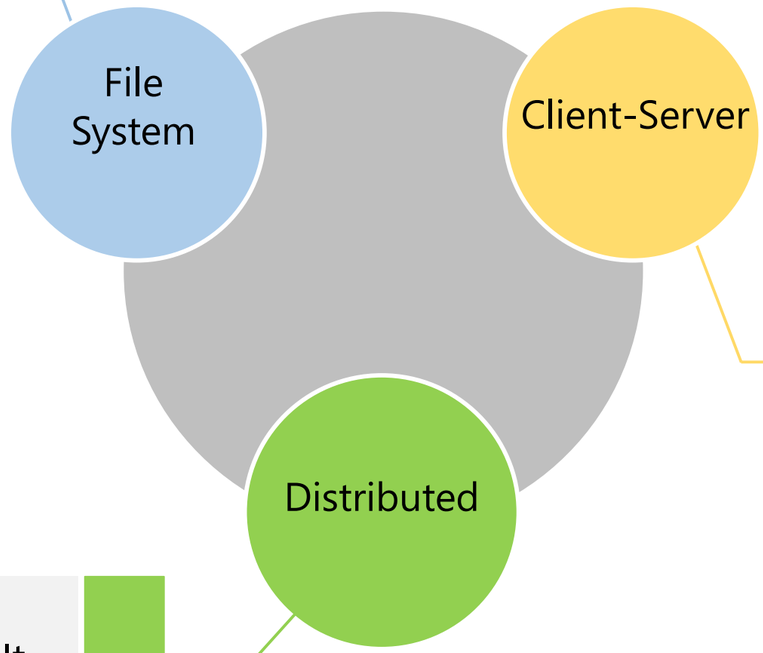
Version Control Systems

Types of Control Systems and Its Supporting Tools

©Simplilearn. All rights reserved.

Types of Source Control Systems

It uses a single repository for each directory. It is often a symbolic link to a shared directory or network drive.



It acts as a central repository which can be used by all. It supports Lock-Modify-Unlock and Copy-Modify-Merge.

It creates replicas of the repository on each computer. It is fast and is based on immutable snapshots of its state.

©Simplilearn. All rights reserved.

File-Based Source Control System

Early source control systems worked at the level of the local file system. It required a subdirectory called SCCS or RCS containing the changes.

Examples of file-based source control systems:

It is a version control system build to scrutinize changes in the files during the development. It takes up a large amount of space and makes it difficult to retrieve information.

SCCS

It is an early version control system with set of Unix commands where multiple developers can develop and maintain documents. Only one user can work at a time.

RCS

©Simplilearn. All rights reserved.

Client-Server Based Source Control System

A central repository can be used by all the developers and others who are involved in the product development and maintenance. An early approach to client-server model was Concurrent Versions System (CVS).

Conflicts occur if two people change a file at the same time. This problem can be solved in two ways:

Lock-Modify-Unlock

- Only one person can change a file at a time
- A file must be locked before changing it
- Different users can't lock a locked file
- Another user must wait for the lock to be released

Copy-Modify-Merge

- Each user has a personal working copy of the file tree
- Changes are made independently and simultaneously
- Private copies are merged into a new version
- All other copies become outdated as soon as a private copy is committed

©Simplilearn. All rights reserved.

Distributed Source Control System

Distributed source control systems create replicas of the repository on each computer. Every user has to work on a replica locally and can do so even being disconnected from the network. They are suited for large projects and independent developers who can work independently and commit the changes for merging.

Examples:

GIT

Bazaar

©Simplilearn. All rights reserved.

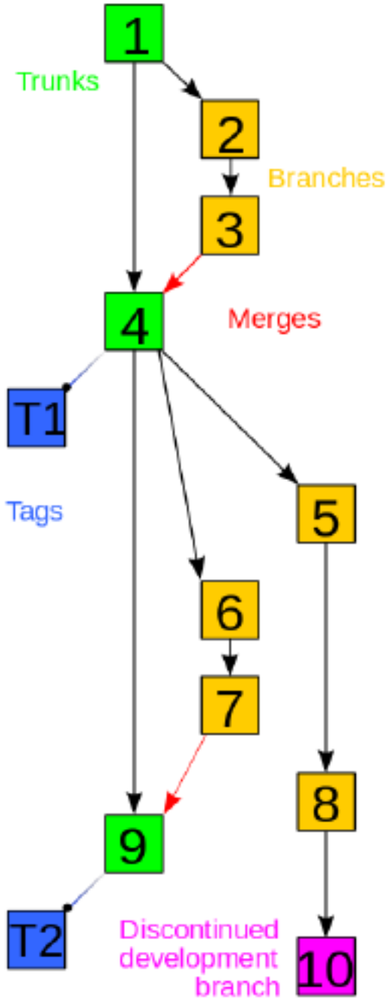
Features of Distributed System

- It is based on immutable snapshots of state with mutable tree structure
- It is conceptually complex, but a user can work offline
- It supports private work where a user can use the changes even for early drafts which need to be published
- It is extremely fast and supports a wide variety of workflows
- It is easy to use once the standards are set, and it allows various development models.

©Simplilearn. All rights reserved.

Repositories Are Key Tools in DevOps

- Repositories are key tools for DevOps, as they are responsible for providing a mechanism to reach stability in an ever-changing environment.
- Repositories can become very complex
- Branches and merges can have intricate relationships
- Graph is a directed acyclic of state changes
- Unused branches should be deleted but ideally nothing is truly deleted.



©Simplilearn. All rights reserved.

Types of Version Control Systems

Git is distributed revision control used to scrutinize the changes in the documents. It is fast, aims for data integrity, and supports nonlinear works. It supports fast branching and merging. Branches are lightweight, and they can perform a commit. Repositories can be published via several protocols such as HTTP, rsync, and FTP.

Bazaar is both, a distributed and a client-server revision control system written in Python for Linux distributions, Mac OS X and Windows variants. It supports collaboration with other revision control Systems such as SVN. It is an innovative tool with flexible features, and it is easier to use.

©Simplilearn. All rights reserved.

Types of Version Control Systems

Subversion, often referred as SVN is a distributed version control tool under Apache license. Developers use subversion to maintain current and previous versions of documents to accomplish its goal as a compatible successor. Path and Revision are the two coordinates which are used to address file system.

Mercurial is a distributed revision control tool which supports Microsoft Windows and Unix-like environments. It is highly scalable, decentralized, and efficient. It is easier to learn, but add-ons should be written in Python. It does not support partial checkouts which is a drawback while working on large projects.

©Simplilearn. All rights reserved.

Types of Version Control Systems

CVS (Concurrent Versions System) is a free client-server revision control system which maintains track of all the work and its changes in a set of documents or files and allows to collaborate. To work well with large text files, it uses delta compression.

Perforce, referred as Perforce Software is used for web-based repository management, team collaboration, and agile planning. It is accepted by larger organizations for better performance on large repositories compared to subversion. Perforce is the right choice if an organization has a large source tree and needs a centralized control system.

©Simplilearn. All rights reserved.

Types of Version Control Systems

RCS

Revision Control System is an early version control system which helps user in creating their own revisions of a document, commit changes, and merge. It operates only on a single file and does not support working with an entire project. Though it is simple to work with, there is minimal security offered.

SCCS

Source Code Control System is a version control system developed to track the status and the changes of the source code and other documents during the development phase. An SCCS file is inclusive of delta table, access, and tracking tags and body of the text. Some of the popular commands are create, edit, delget, get, and prt.

©Simplilearn. All rights reserved.

Tools Supporting Version Control Systems

Monotone is an open-source software tool for distributed revision control. It helps in tracking revision of codes, documents and also tracks the history across renames. It strongly supports a diverge/merge workflow. Its noteworthy features are good support for internationalization and localization, very low maintenance, and stable GUI. It does not support potential users to checkout from a proxy network due non-HTTP protocol.

Plastic SCM is a cross-platform commercial software tool for distributed version control. It is a full version control stack which includes a command-line tool, a GUI, different merge tools, and a large number of IDEs. Its popular features are built-in 3-way merge, directory versioning, distributed and centralized operations. It supports popular operating systems such as Mac OS X, Windows, and few Linux platforms.

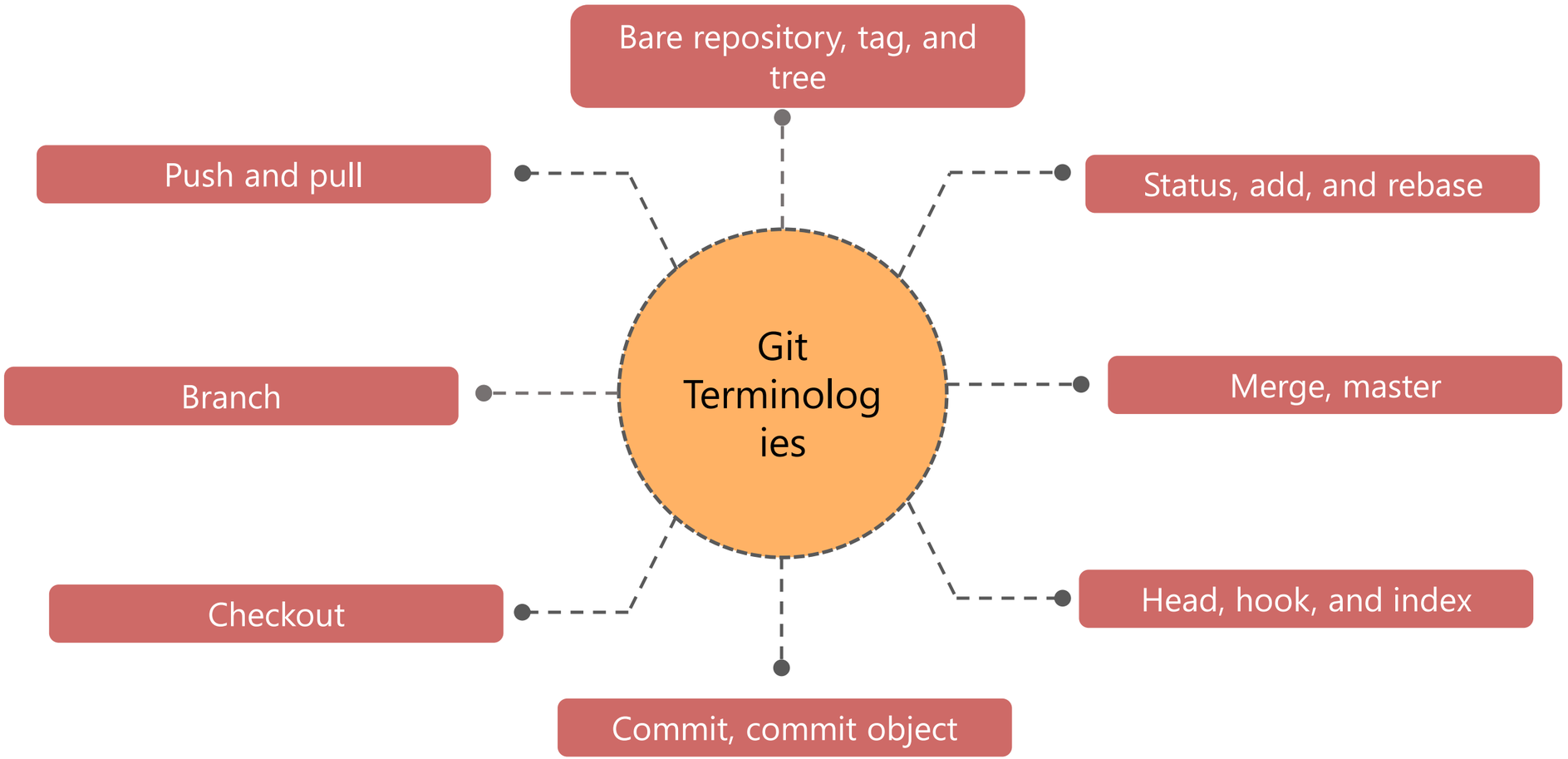
©Simplilearn. All rights reserved.

Version Control Systems

Overview of Git

©Simplilearn. All rights reserved.

Popular Terminologies



©Simplilearn. All rights reserved.

Git Workflow

1

git init

2

git status

3

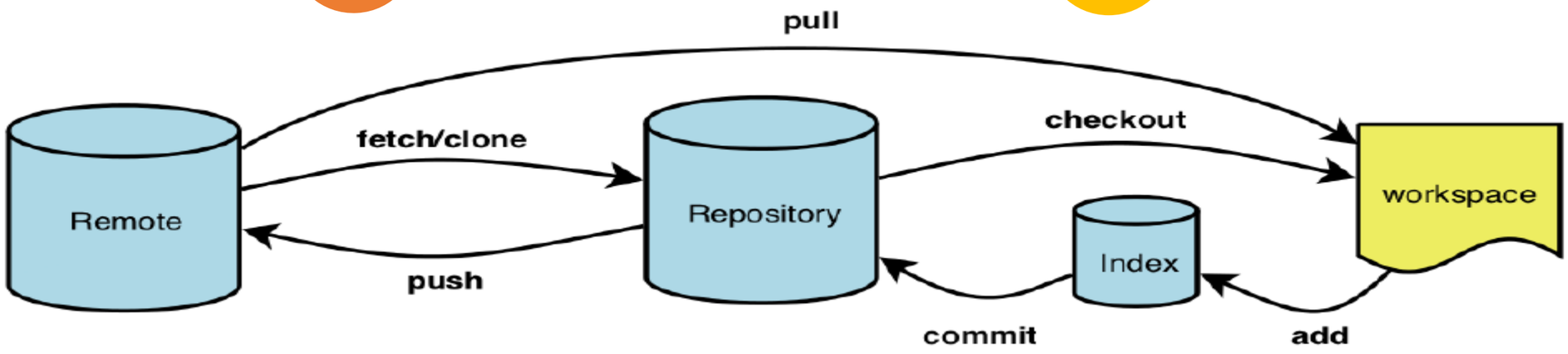
git add <files>
git commit <files>

4

git push origin master

5

git rebase <parameters>
git checkout <references>
git merge <references>



©Simplilearn. All rights reserved.

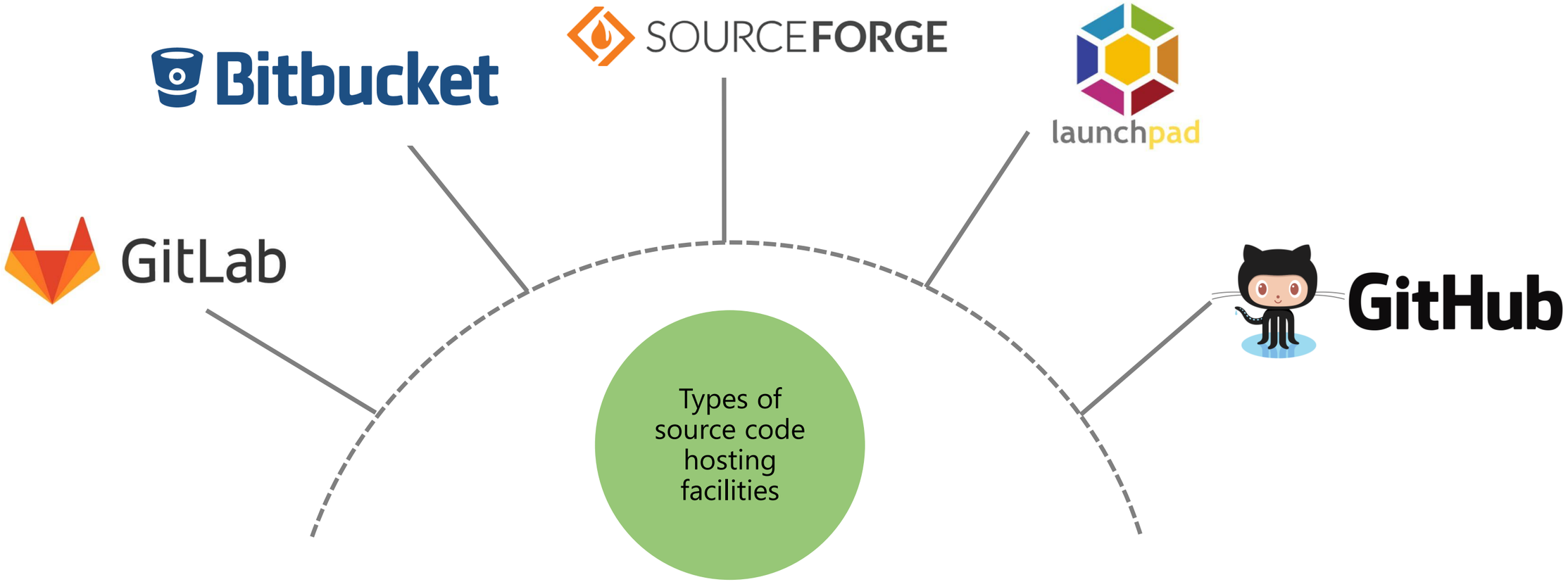
Version Control Systems

Overview of Source Code and Version Control Hosts

©Simplilearn. All rights reserved.

Source Code Hosting Facility

A source code repository is a web hosting and file archive facility where larger data such as source code is stored either privately or publicly. The most popular source code hosting sites are listed:



©Simplilearn. All rights reserved.

Supported Features

Name	Code Review	Bug Tracking	Web Hosting	Build System
GitLab	Yes	Yes	Yes	Yes
Bitbucket	Yes	Yes	Yes	Yes
SourceForge	Yes	Yes	Yes	No
launchpad	Yes	Yes	No	Yes (Only Ubuntu)
GitHub	Yes	Yes	Yes	Via third party

©Simplilearn. All rights reserved.

Supported Version Control Systems

Name	CVS	Git	SVN	Perforce
GitLab	No	Yes	No	No
Bitbucket	No	Yes	No	No
SourceForge	Dropped	Yes	Yes	-
launchpad	Import Only	Yes	Import Only	No
GitHub	No	Yes	Partial	No

©Simplilearn. All rights reserved.

Popularity and the Community Usage

Name	Users	Projects
GitLab	100,000	546,000
Bitbucket	5,000,000	Private
SourceForge	3,700,000	500,000
launchpad	3,965,288	40,881
GitHub	24,000,000	69,000,000

©Simplilearn. All rights reserved.

Assisted Practice

Duration: 40 mins

Deploy the Files to GitHub via Git.

Problem Statement: You are given a project to add all the necessary files to your GitHub account via Git interactive terminal.

Access: Click on the **Labs** tab on the left side panel of the LMS. Copy or note the username and password that is generated. Click on the **Launch Lab** button. On the page that appears, enter the username and password in the respective fields, and click **Login**.

©Simplilearn. All rights reserved.

Assisted Practice: Guidelines to Deploy Files to Hosting Servers

1. List the basic structure of Git, GitHub, and Bitbucket.
2. Create files with every extension, and add random content in it (.txt, .html, .css, .ts, .js, .java, .c).
3. Initialize the Git repo locally and commit the files.
4. Make changes in random files, and repeat step 2.
5. Sign in to the GitHub account. Create an account if it is not created.
6. Connect the local and remote repositories.
7. Add the codes to the repositories of GitHub.
8. Repeat this demonstration three times where each time the commit should be for different repositories.

©Simplilearn. All rights reserved.

Key Takeaways

You are now able to:

- Decide the suitable version control systems for your organization
- Explain the role of version control systems
- Identify the types of control systems and the supporting tools
- Set up Git
- Identify the suitable source code and version control hosts

©Simplilearn. All rights reserved.

Knowledge Check

©Simplilearn. All rights reserved.

Knowledge Check 1	**Choose the software tool that supports Version Control System.**

a. Perforce

b. Plastic SCM

c. Git

d. CVS

©Simplilearn. All rights reserved.

Knowledge Check 1	**Choose the software tool that supports Version Control System.**

a. Perforce

b. Plastic SCM

c. Git

d. CVS

The correct answer is **b. Plastic SCM**

©Simplilearn. All rights reserved.

Knowledge Check 2	**Choose the correct File-based version control system.**

a. SCCS and RCS

b. Perforce and SVN

c. Git and SCCS

d. SCCS , RCS, and Git

©Simplilearn. All rights reserved.

Knowledge Check 2	**Choose the correct File-based version control system.**

a. SCCS and RCS

b. Perforce and SVN

c. Git and SCCS

d. SCCS , RCS, and Git

The correct answer is **a. SCCS and RCS**

©Simplilearn. All rights reserved.

Knowledge Check 3	**Choose the correct solutions for the conflict raised in client-server based systems.**

a. Lock-Modify-Unlock

b. Copy-Modify-Merge

c. Create-Modify-Merge

d. Options a and b

©Simplilearn. All rights reserved.

Knowledge Check 3	**Choose the correct solutions for the conflict raised in client-server based systems.**

a. Lock-Modify-Unlock

b. Copy-Modify-Merge

c. Create-Modify-Merge

d. Options a and b

The correct answer is **d. Options a and b**

©Simplilearn. All rights reserved.

Knowledge Check 4	**A built-in 3-way merge is a popular feature of _____**

a. Bazaar

b. Plastic SCM

c. Perforce

d. Git

©Simplilearn. All rights reserved.

Knowledge Check 4	**A built-in 3-way merge is a popular feature of _____**

a. Bazaar

b. Plastic SCM

c. Perforce

d. Git

The correct answer is **b. Plastic SCM**

©Simplilearn. All rights reserved.

Knowledge Check 5	**Which among the following does not support the build system?**

a. SourceForge

b. GitLab

c. GitHub

d. None of the above

©Simplilearn. All rights reserved.

Knowledge Check 5	**Which among the following does not support the build system?**

a. SourceForge

b. GitLab

c. GitHub

d. None of the above

The correct answer is **a. SourceForge**

©Simplilearn. All rights reserved.

Lesson-End Project

Deploy the Files to Bitbucket via Git.

Duration: 40 mins

Problem Statement:

Create a directory named “Lesson-02” and perform the following:

- Create multiple files with .html, .xml, .txt, .py, .js, and .java.
- Add random content in each file.
- Connect git local to Bitbucket remote.
- Push the directory “Lesson-02” to Bitbucket account.

Access: Click on the **Labs** tab on the left side panel of the LMS. Copy or note the username and password that is generated. Click on the **Launch Lab** button. On the page that appears, enter the username and password in the respective fields, and click **Login**.

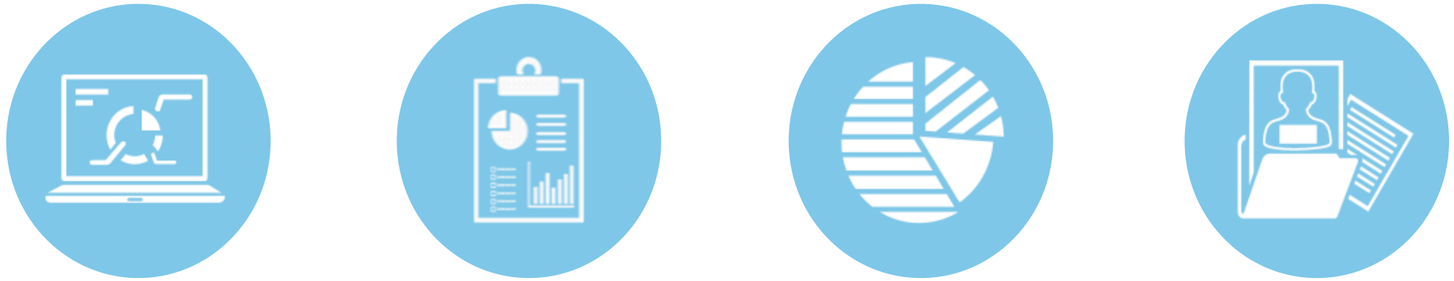
©Simplilearn. All rights reserved.

Thank You

©Simplilearn. All rights reserved.

DevOps Certification Training

Lesson 03: Continuous Integration, Continuous Deployment, and Build Tools



© Simplilearn. All rights reserved.

Learning Objectives

By the end of this lesson, you will be able to:

- Describe the importance of continuous integration and continuous deployment
- List the features of Jenkins and demonstrate their uses
- List the features of TeamCity and demonstrate their uses
- Select a suitable build tool for your organization

©Simplilearn. All rights reserved.

Continuous Integration, Continuous Deployment, and Build Tools

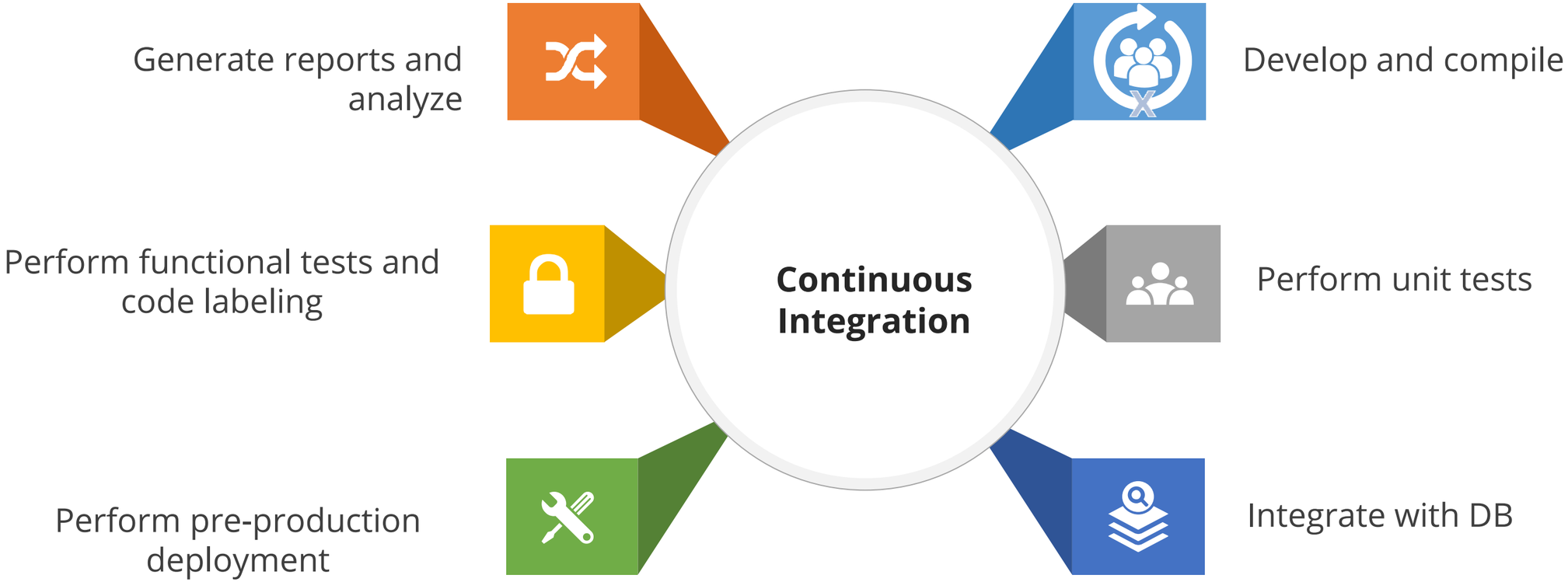
Overview and Importance of Continuous Integration and Continuous Deployment

©Simplilearn. All rights reserved.

Overview of Continuous Integration

Continuous integration is a development practice of code integration into a shared repository.

Each integration is verified by an automated build and automated tests.

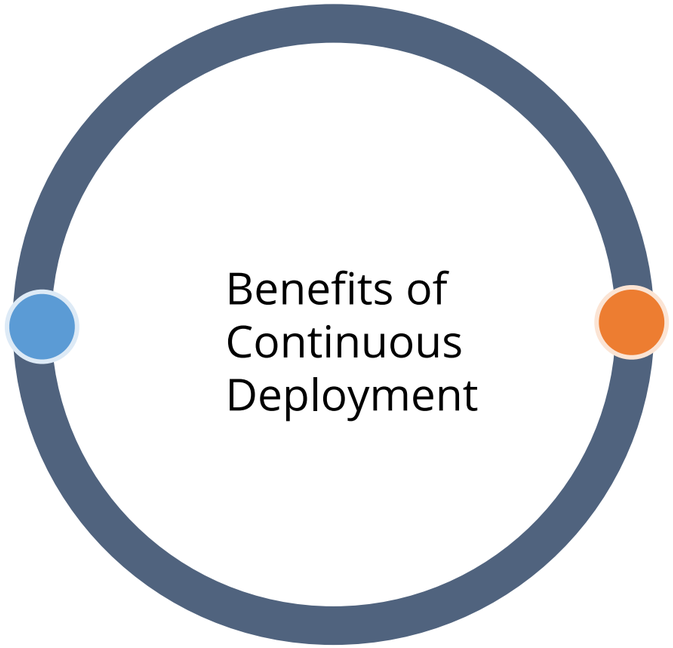


©Simplilearn. All rights reserved.

Overview of Continuous Deployment

Continuous Deployment is an extension of continuous integration. It targets to reduce the time between development team writing one new line of code and using it in production.

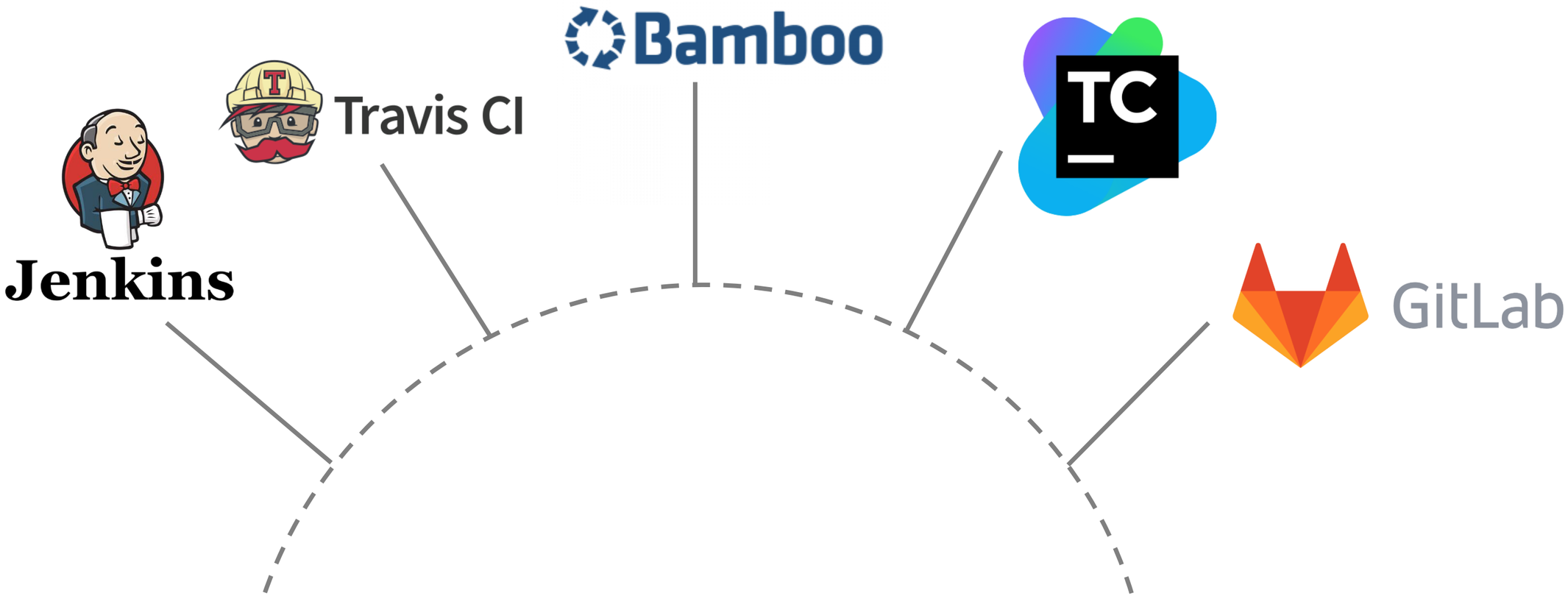
Faster return on investment for each feature as it gets developed



Faster feedback from end users on each new feature as it is released to production

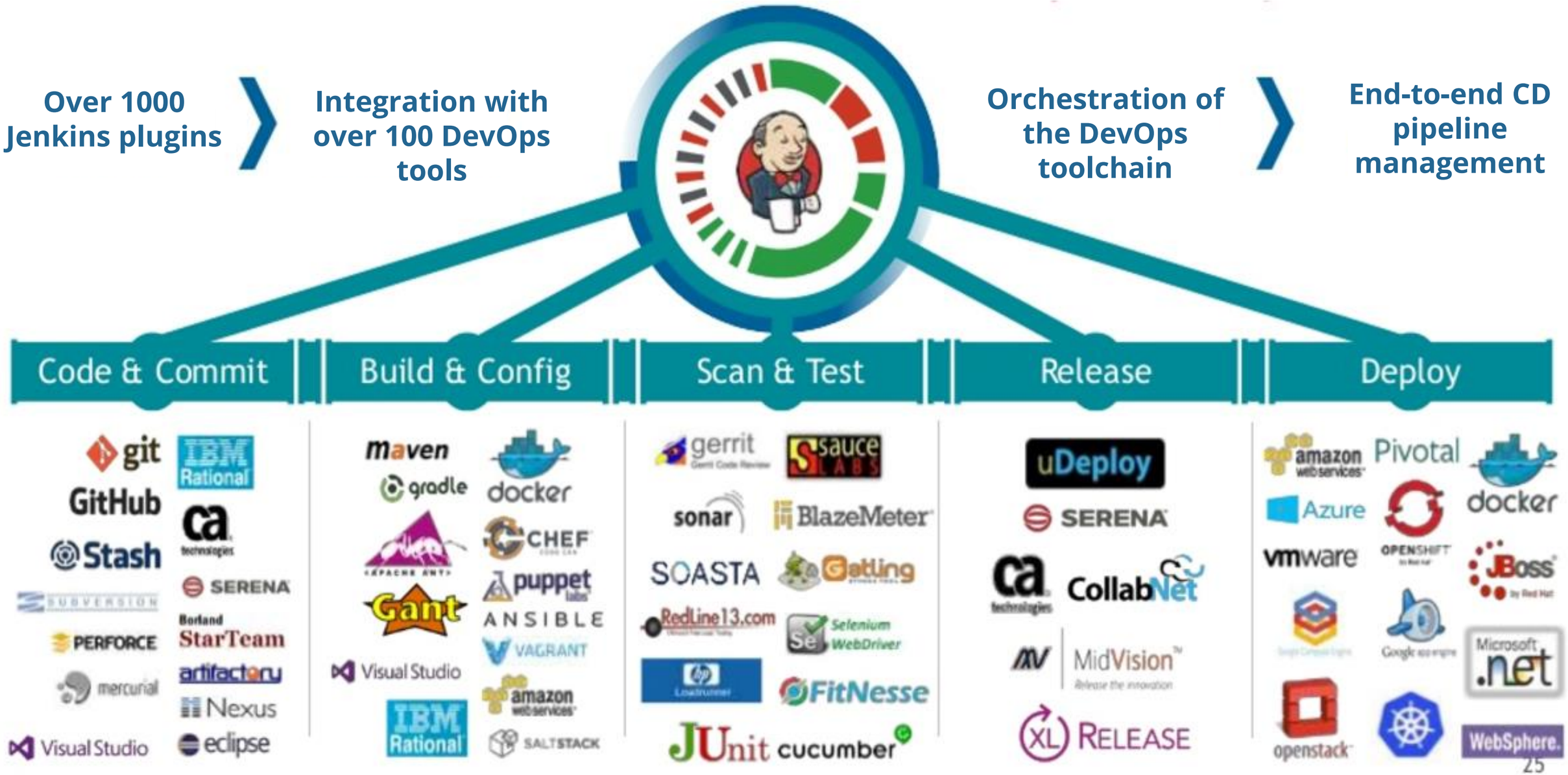
©Simplilearn. All rights reserved.

Popular Tools in Continuous Integration and Continuous Deployment



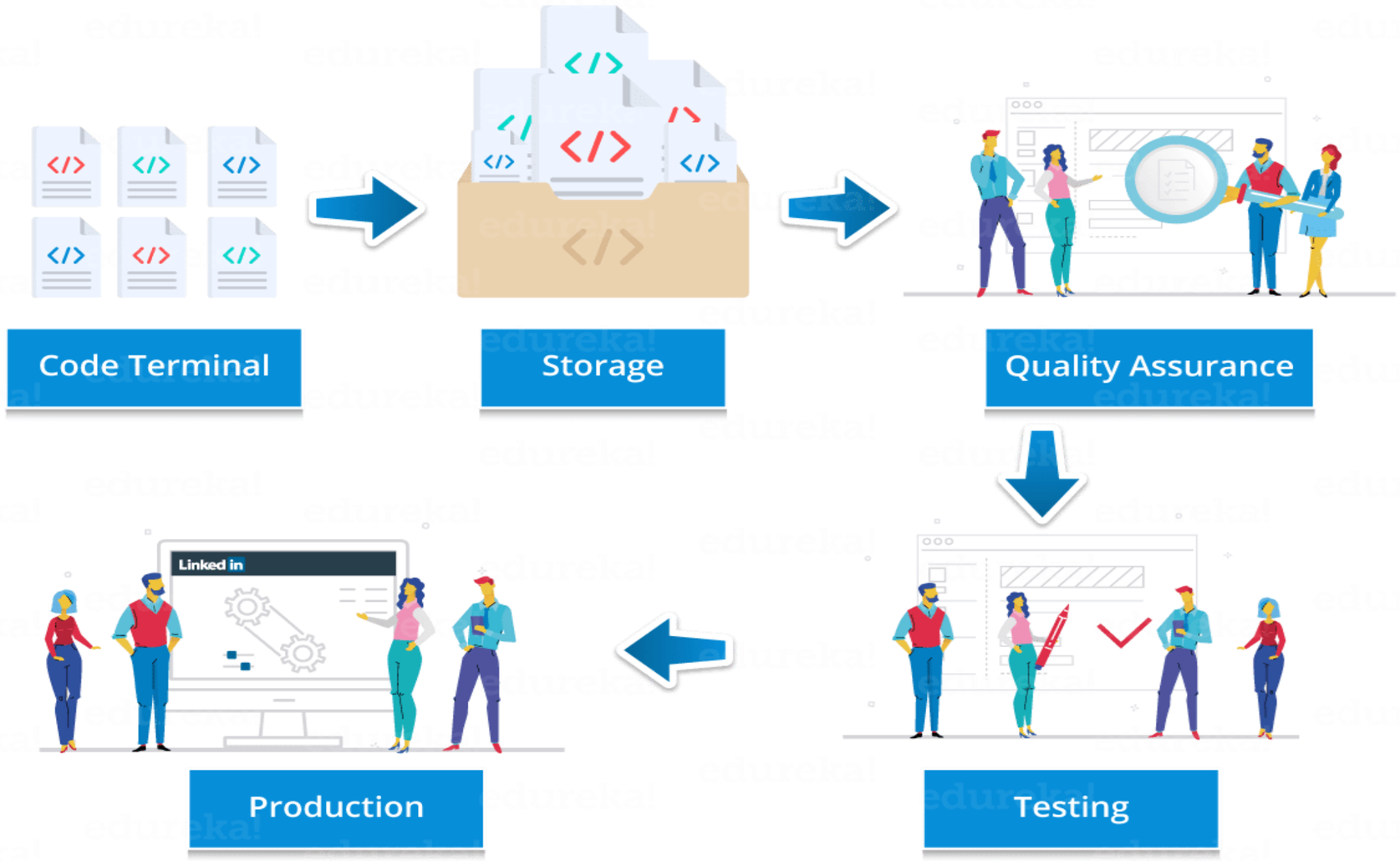
©Simplilearn. All rights reserved.

Continuous Integration with Jenkins



©Simplilearn. All rights reserved.

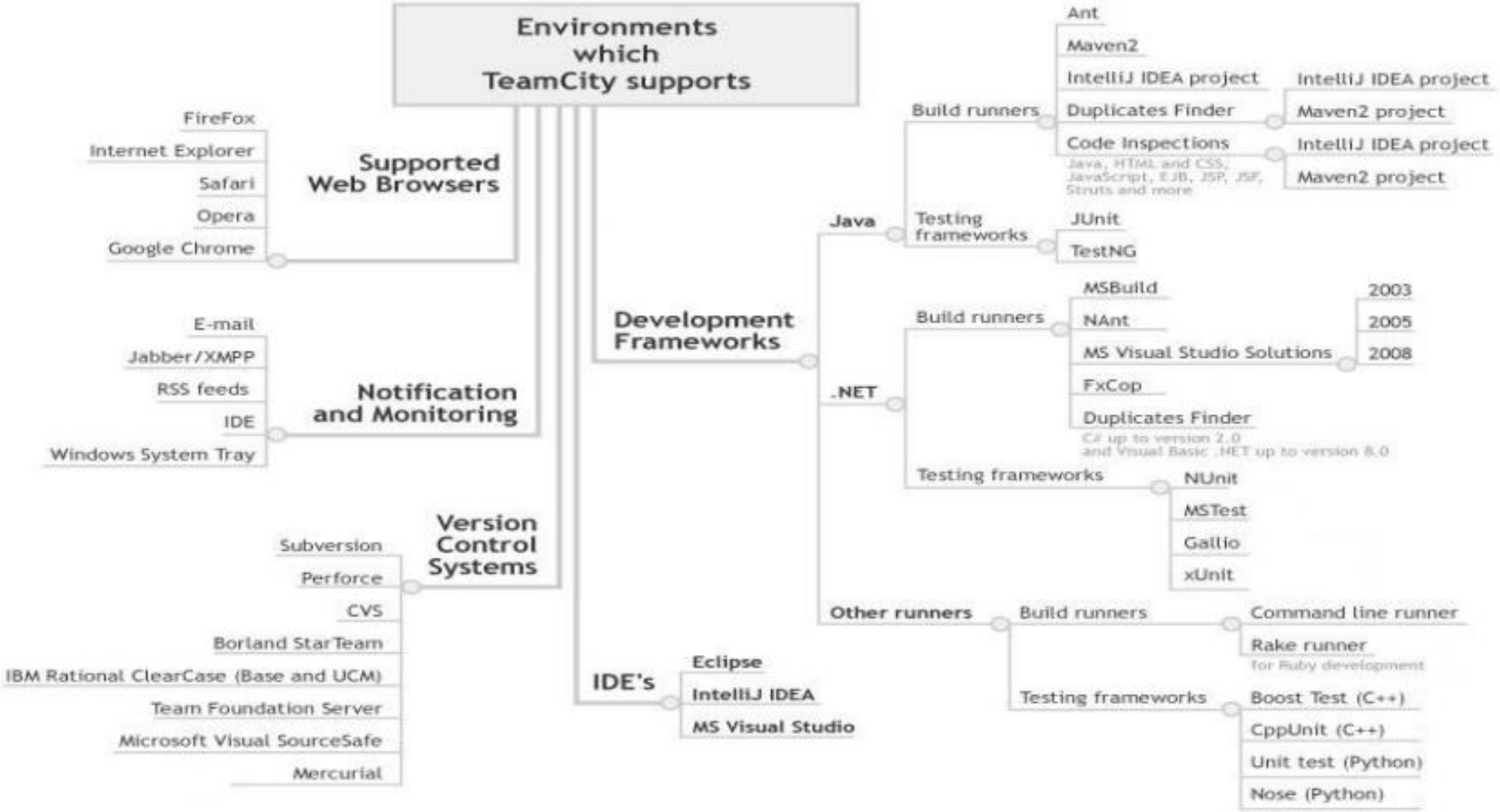
Continuous Deployment with Jenkins



A case study by LinkedIn

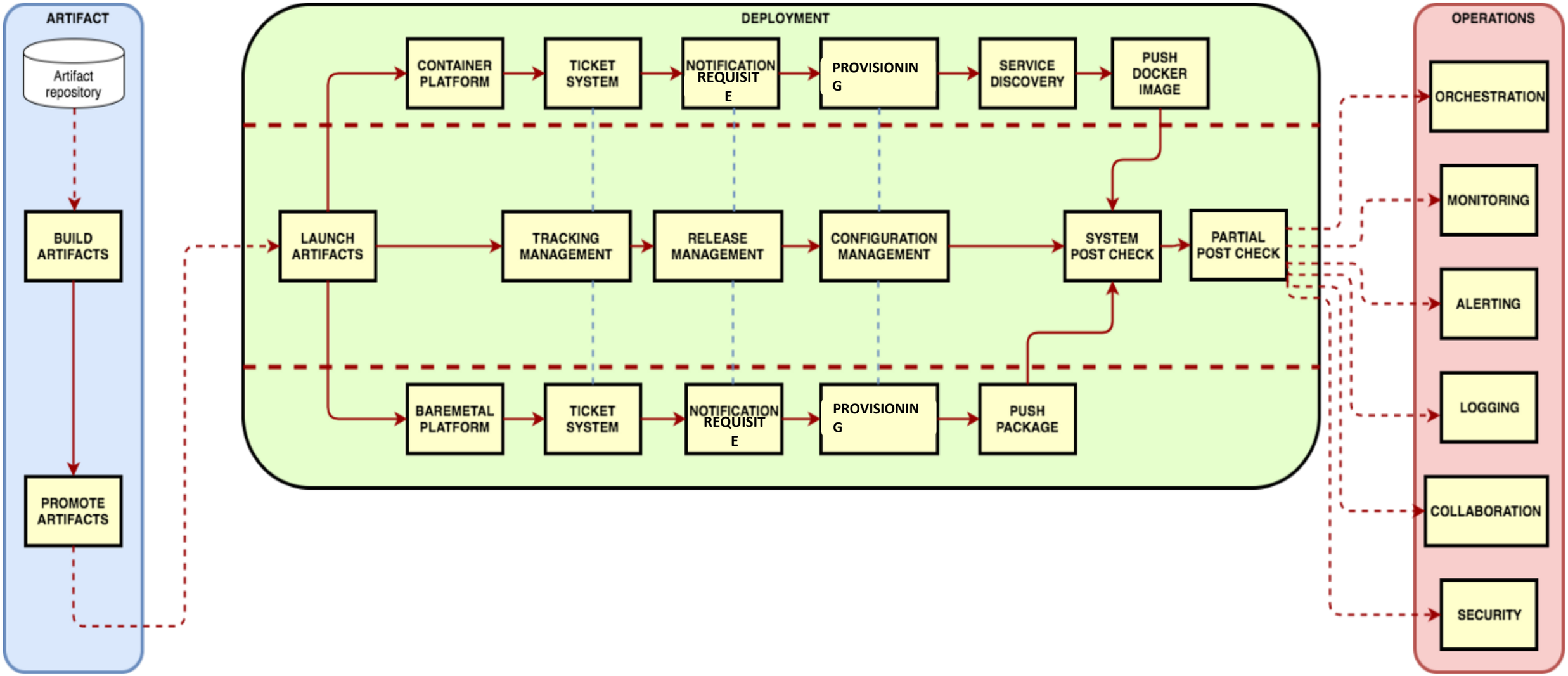
©Simplilearn. All rights reserved.

Continuous Integration with TeamCity



©Simplilearn. All rights reserved.

Continuous Deployment with TeamCity



©Simplilearn. All rights reserved.

Continuous Integration, Continuous Deployment, and Build Tools

Overview and Features of Jenkins

©Simplilearn. All rights reserved.

simplilearn

Jenkins as a Continuous Integration Tool

Jenkins is a Java-based, open source automation tool. It functions as a server and is a software development and cross-platform tool used for continuous integration and continuous deployment.

CI server

It can be used as a CI server or as a continuous delivery hub for a project

Distribution

It can easily distribute work across different machines and help trigger builds, tests, and deployments to multiple machines and platforms faster

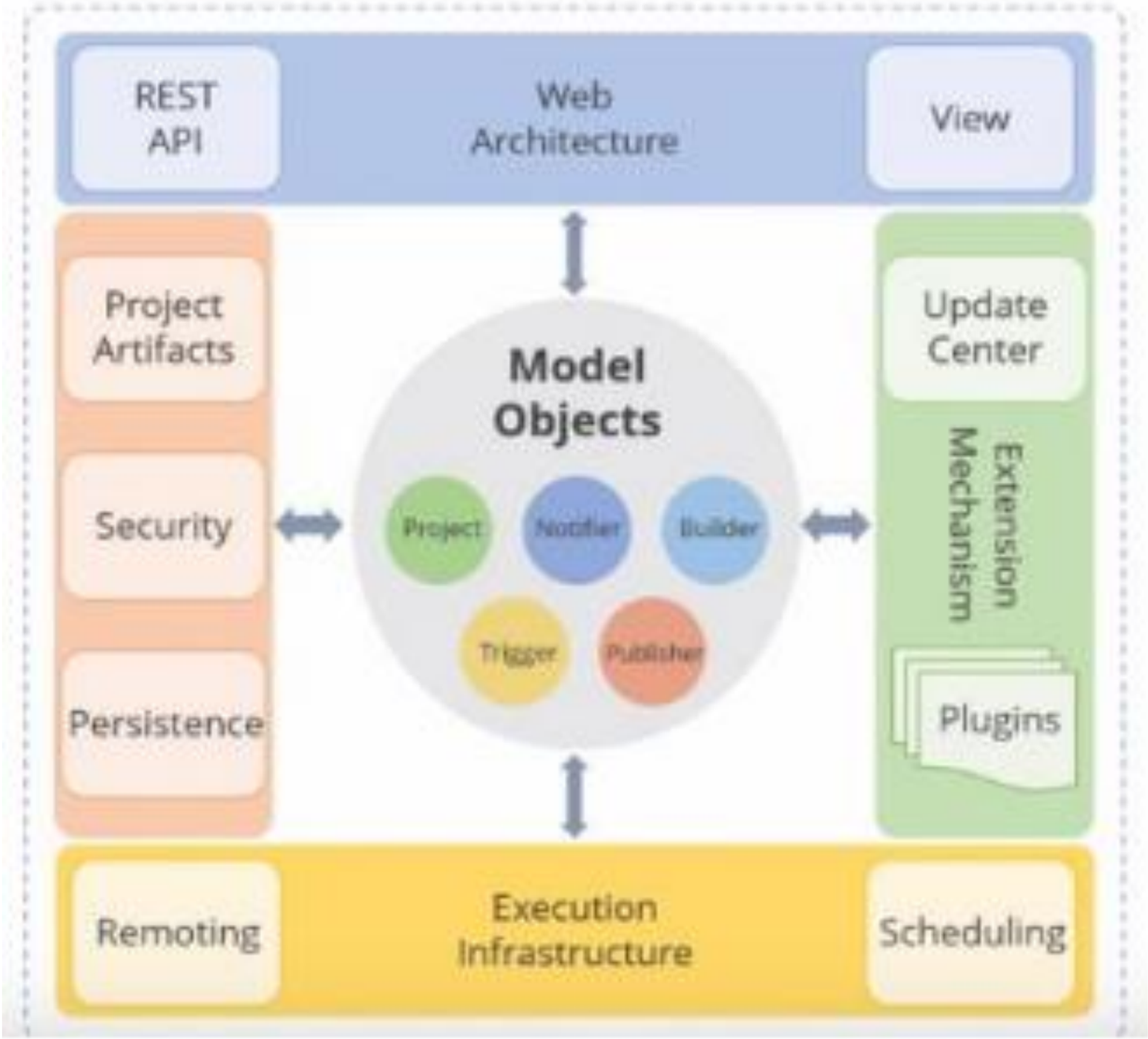
Cross-platform

It works on iOS, .Net, Android Development, Ruby, and Java

©Simplilearn. All rights reserved.

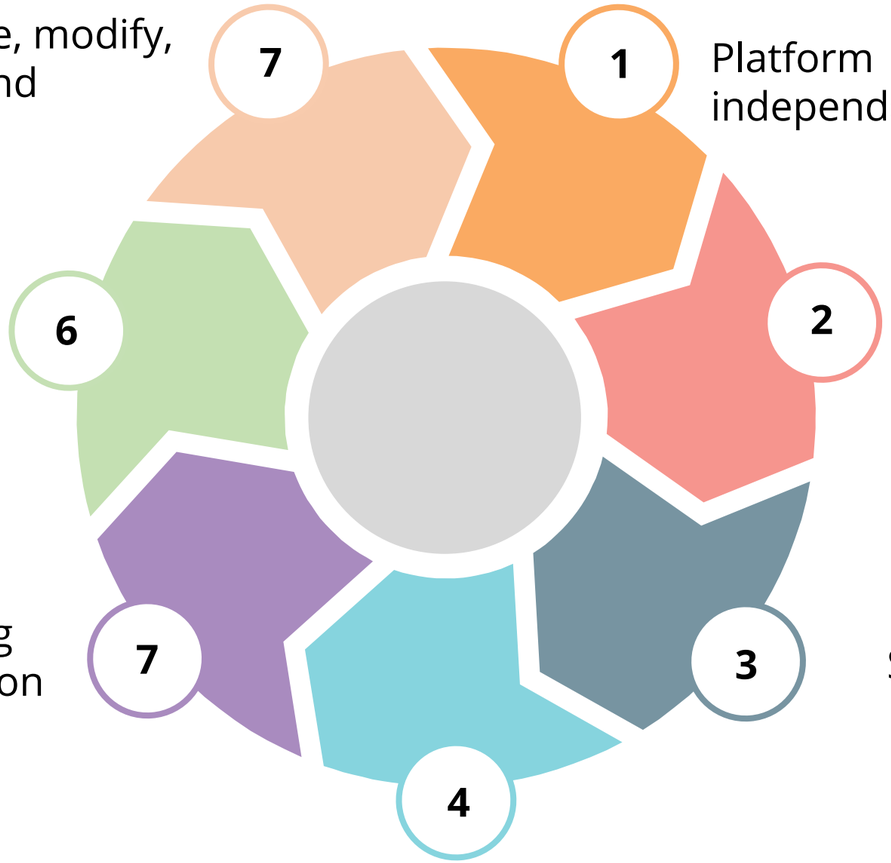
Architecture of Jenkins

- Jenkins has classes like project and build.
- It uses Jelly as the view technology.
- It uses file system to store its data. Directories are created inside $JENKINS_HOME.
- It supports plugins, which can plug into those extension points and extend the capabilities of Jenkins.



©Simplilearn. All rights reserved.

Popular Features of Jenkins



1. Platform independent
2. Rich plugin ecosystem
3. Support from large communities
4. Scaling of large error-ridden integrations
7. Automation integration enabling immediate detection and resolution of issues
6. Open source and user-friendly
7. Easy to configure, modify, and extend

©Simplilearn. All rights reserved.

Build Status and Job Health

Status of the build	Description
	Failed
	Unstable
	Success
	Pending
	Disabled
	Aborted

Figure a: Build status

Job health	Description
	No recent builds failed
	20-40% of recent builds failed
	40-60% of recent builds failed
	60-80% of recent builds failed
	All recent builds failed
	Unknown status

Figure b: Weather reports

©Simplilearn. All rights reserved.

Assisted Practice

Duration: 30 mins

Set up Jenkins

Problem Statement: You are given a project to install and configure Jenkins on your Ubuntu operating system.

Access: Click on the **Labs** tab on the left side panel of the LMS. Copy or note the username and password that is generated. Click on the **Launch Lab** button. On the page that appears, enter the username and password in the respective fields, and click **Login**.

©Simplilearn. All rights reserved.

Assisted Practice: Guidelines to Install and Configure Jenkins

1. Login to your Ubuntu lab provided with the course.
2. Open the terminal and execute the command available in the lab document 3.1 to add the key to the system.
3. Edit the sources.list file, add the command to the file, and save it.
4. Update the apt-get package.
5. Install JDK 8+ version.
6. Install Jenkins via apt-get package.
7. Navigate to x.x.x.x:8080 in the browser of your virtual Machine.
8. Get the password and enter it in the Jenkins window.
9. Create a new role/job in Jenkins.
10. Explore the free style project section and build section.

©Simplilearn. All rights reserved.

Continuous Integration, Continuous Deployment, and Build Tools

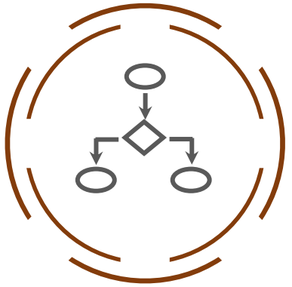
Overview and the Features of TeamCity

©Simplilearn. All rights reserved.

simplilearn

TeamCity as a Continuous Integration Tool

TeamCity is a Java-based, management and continuous integration server. It is a licensed commercial software used for continuous integration and continuous deployment.



Gated commits

Build grid

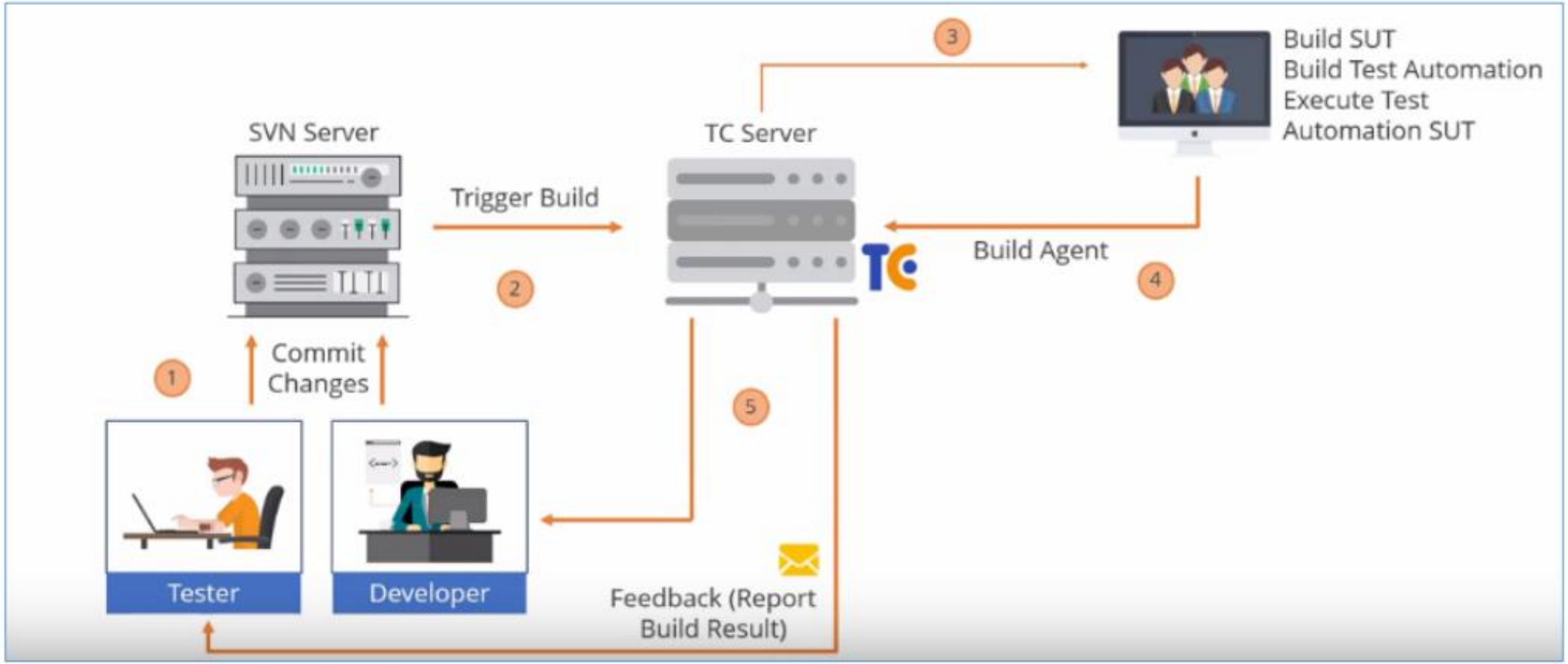
Integrated code

Integration with IDEs

Cross-platform support

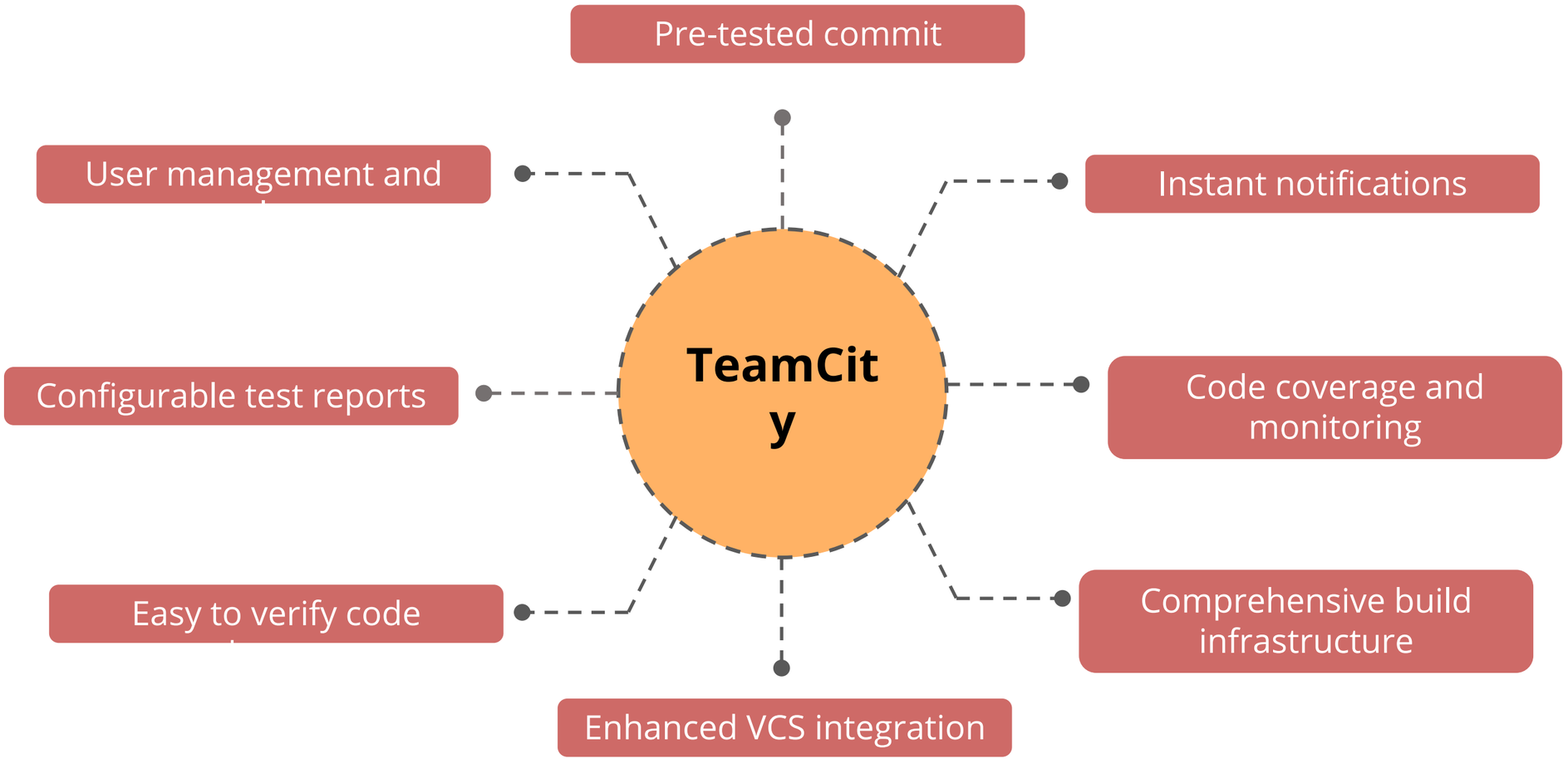
©Simplilearn. All rights reserved.

TeamCity Workflow



©Simplilearn. All rights reserved.

Popular Features of TeamCity



©Simplilearn. All rights reserved.

Continuous Integration, Continuous Deployment, and Build Tools

Build Tools and Their Uses

©Simplilearn. All rights reserved.

Build Tools

Build tools are programs that automate the creation of executable applications from the source code. Automation tools allow the build process to be more consistent.

1

Make

Software

2

BASH

THE BOURNE-AGAIN SHELL

#!/bin/bash

3

<APACHE ANT>

4

Apache Maven

http://maven.apache.org/

5

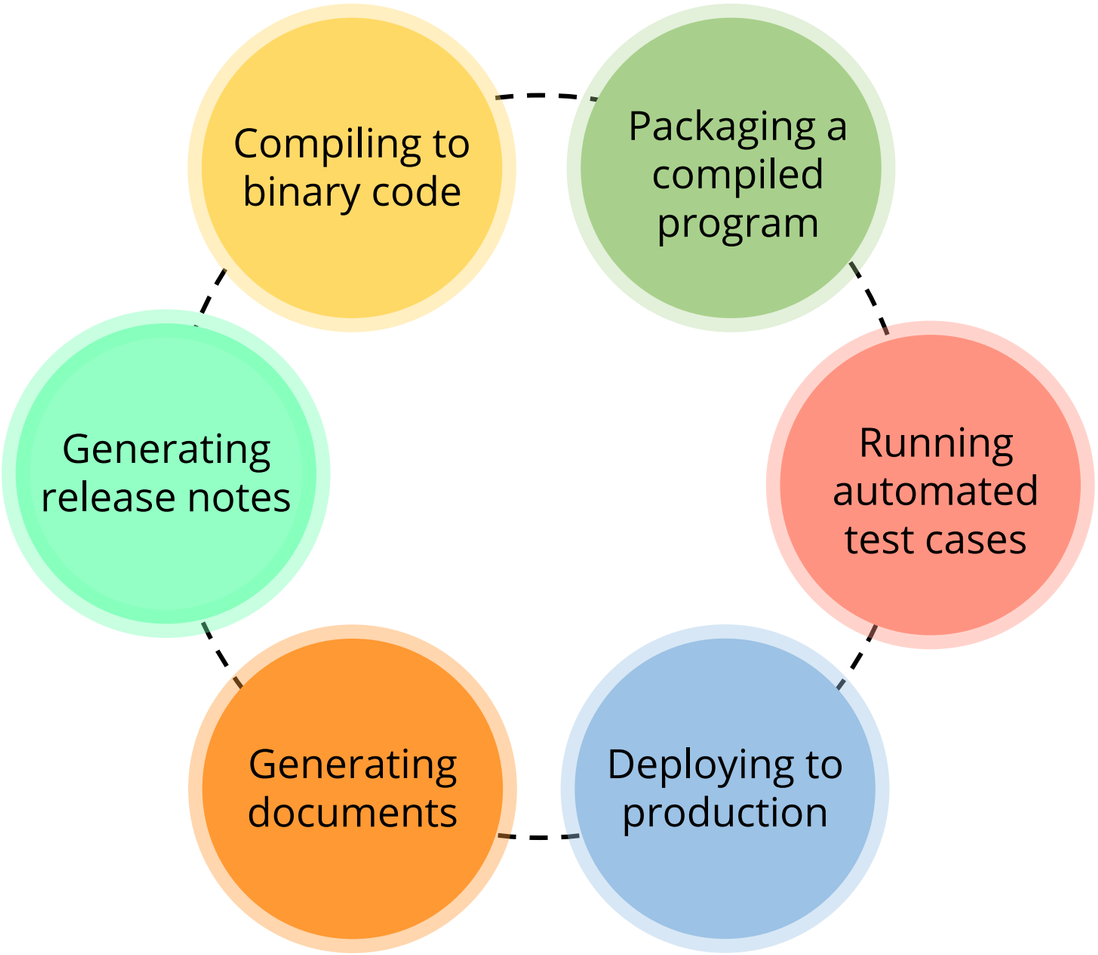
GRUNT

6

Gulp

©Simplilearn. All rights reserved.

Popular Features of Build Tools



©Simplilearn. All rights reserved.

Overview of Apache Ant

Apache Ant is a Java library and a command-line tool. It aims to drive processes described in build files as targets and extension points dependent on each other.

Features of

- Supplies a number of built-in tasks allowing to compile, test, and run
- Flexible and does not impose coding conventions
- Pilot any type of process
- Build solution combining build tool and dependency with Apache Ivy
- Users can develop custom “Antlibs” using Java
- Solves Make’s portability problems

©Simplilearn. All rights reserved.

Limitations of Apache Ant

04

Ant build files are complex and verbose as they are hierarchical and partly ordered

01

Undefined properties are not raised as errors but left as unexpanded reference

03

Older tasks use default value, which are not consistent, and changing defaults would break existing Ant scripts

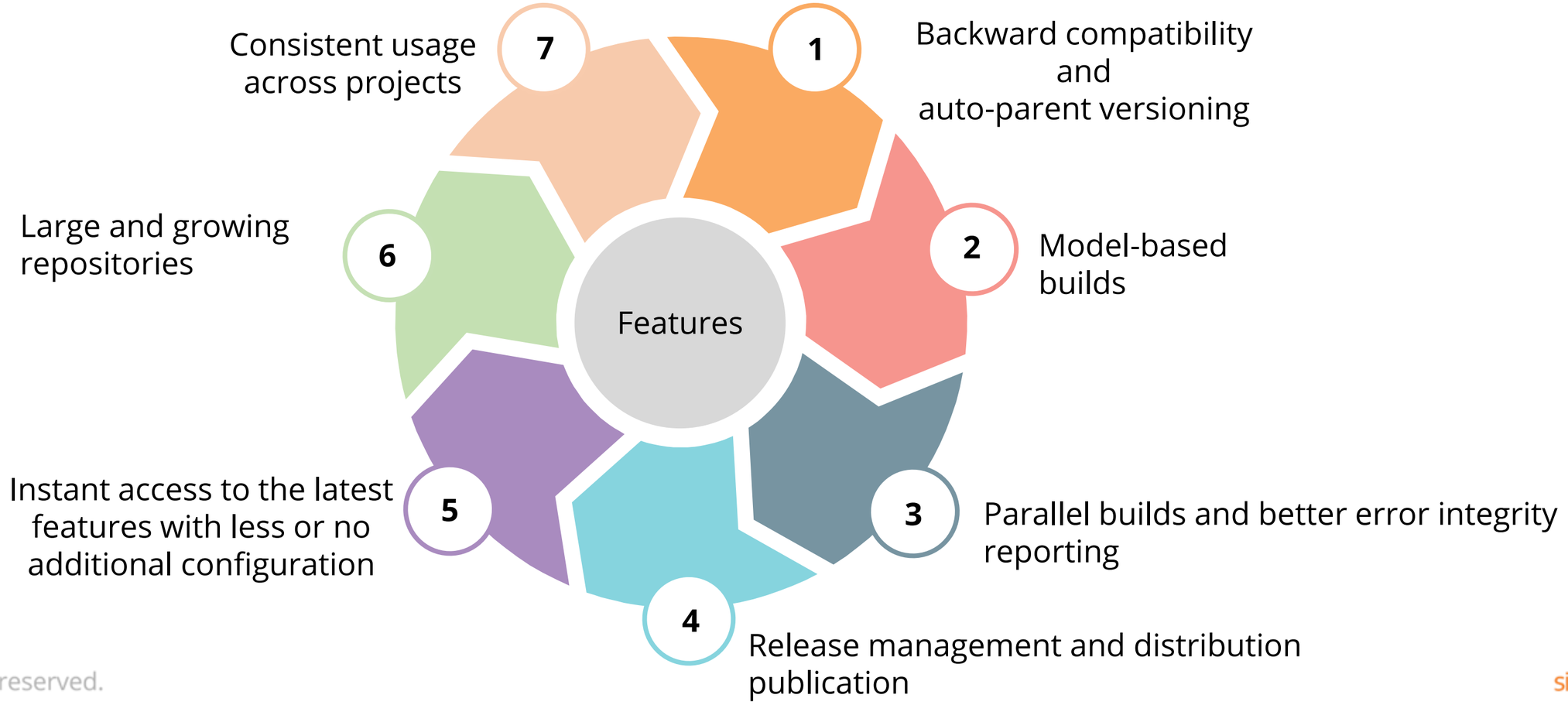
02

Ant has limited fault handling rules and lazy property evaluation is not supported

©Simplilearn. All rights reserved.

Overview of Maven

Apache Maven is a software project management and comprehension tool. Based on Project Object Model (POM), Maven can manage a project's build reporting and documentation from a central piece of information.

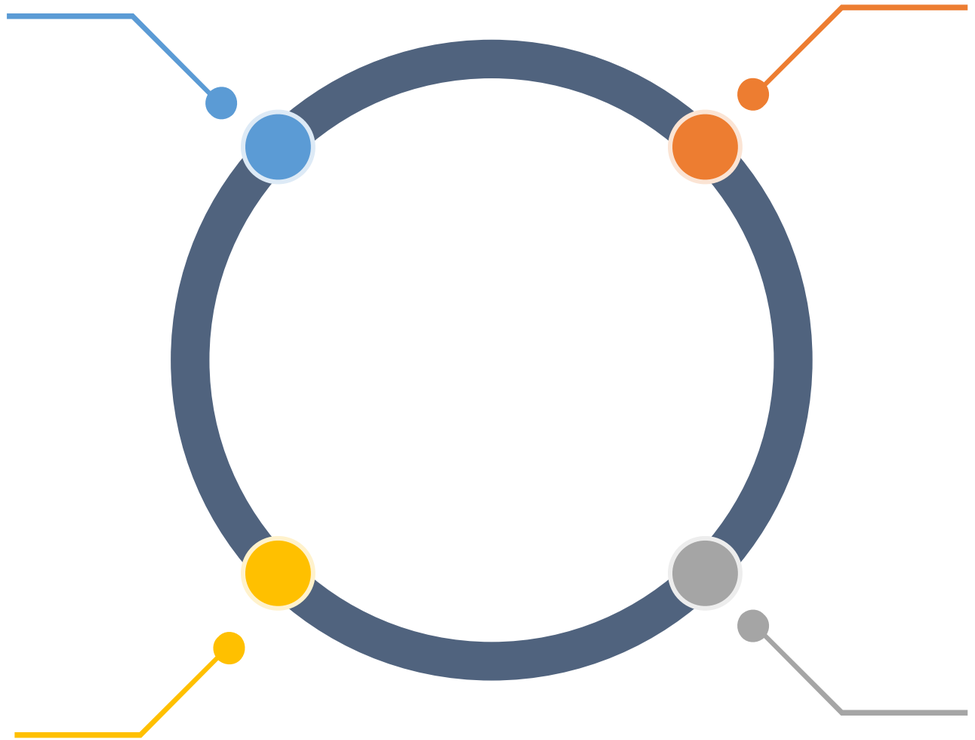


©Simplilearn. All rights reserved.

Drawbacks of Maven

Unreliable with Eclipse

Unable to depend on the outcome status



Slow and a partial black box

Verbose and complex

©Simplilearn. All rights reserved.

Maven over Ant

Better collaboration and debugging

More powerful builds

More componentized builds and reduced duplication

Better dependency management

Apache Maven
http://maven.apache.org/

More consistent project structure

©Simplilearn. All rights reserved.

Project Object Model (POM)

Project Object Model is an XML representation of a Maven project, which provides general configurations such as project's name, its owner, and its dependencies on other projects.

- The POM needs to define the Group ID, Artifact ID, and Version
- The packaging should also be declared – the default is jar

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
                       http://maven.apache.org/xsd/maven-4.0.0.xsd">

   <modelVersion>4.0.0</modelVersion>
   <groupId>eu.totaleclipse</groupId>
   <artifactId>calculator</artifactId>
   <version>1.0</version>
   <packaging>jar</packaging>
</project>
```

©Simplilearn. All rights reserved.

Overview of Grunt

Grunt is a JavaScript-based task runner, which is used to automate repetitive tasks in a workflow. It can be used as a command-line tool for JavaScript objects.

- Eases workflow such as writing a set up file
- Helps in automation of repetitive task with less effort
- Minifies the files such as .html, .CSS
- Includes built-in tasks to extend functionality of plugins
- Speeds the development flow and enhances performance
- Supports small infrastructure which is a best fit for new codebase
- Aims at reducing the chances of errors during repetitive tasks
- Currently, it has over 4000 plugins and can be used in larger production sites

©Simplilearn. All rights reserved.

Overview of Gulp

Gulp is an open-source JavaScript toolkit used as a streaming build system in front-end web development. It automates time-consuming and repetitive tasks involved in development.

Features

- Code minification and concatenation
- Usage of pure JavaScript code
- Converts LESS or SASS to CSS compilation
- Manages file manipulation in the memory

Advantages

- Easy to code
- Easy to test the web apps
- Plugins are simple to use

Disadvantages

- More number of dependencies
- Multiple tasks cannot be performed
- Configuration is tedious

©Simplilearn. All rights reserved.

Assisted Practice

Duration: 50 mins

Continuous Integration with Jenkins and Maven

Problem Statement: You are given a project to configure Jenkins, poll Git commits and build the project code using Maven on your Ubuntu operating system.

Access: Click on the **Labs** tab on the left side panel of the LMS. Copy or note the username and password that is generated. Click on the **Launch Lab** button. On the page that appears, enter the username and password in the respective fields, and click **Login**.

©Simplilearn. All rights reserved.

Assisted Practice: Guidelines to Configure Jenkins and Maven

1. Login to your Ubuntu lab provided with the course.
2. Login to Jenkins and create the first Jenkins job.
3. Install and configure Maven.
4. Configure Jenkins with Java, Git, and Maven.
5. Create a Jenkins job for your maven build project and run the project.
6. Poll Git for commits and automatically trigger the build.
7. Build the trigger using Push mechanism instead of Pull.
8. Repeat steps 6 and 7 multiple times to observe the results at **console output** section.

©Simplilearn. All rights reserved.

Continuous Integration, Continuous Deployment, and Build Tools

Working with Jenkins File

©Simplilearn. All rights reserved.

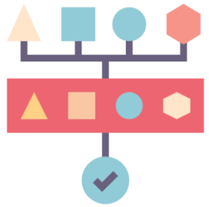
Jenkinsfile

- The Pipeline definition in the **Jenkinsfile** can be committed to the repository along with the source code.
- This method is more consistent as the Pipeline structure becomes strongly tailored to the project.

For example: if you don't need the code compilation because your programming language is interpreted, then you won't have the Compile stage.

©Simplilearn. All rights reserved.

Creating Jenkinsfile



A Jenkinsfile for a commit will have almost the same content as the commit Pipeline.

The only difference will be that the checkout stage becomes redundant because Jenkins has to checkout the code as well as the Jenkinsfile and then read the Pipeline structure from the Jenkinsfile.

Jenkins needs to know the repository address before it reads Jenkinsfile.

Once the Jenkinsfile is created, you can push it to the GitHub repository.

©Simplilearn. All rights reserved.

Creating Jenkinsfile

Here is a sample Jenkinsfile for a commit Pipeline:

```
Pipeline {
    agent any stages {
        stage("Compile") {
            steps {
                sh "./gradlew compileJava"
                }
            }
        stage("Unit test") {
            steps {
                sh "./gradlew test"
                }
            }
        }
    }
```

©Simplilearn. All rights reserved.

Running Jenkinsfile

After the Jenkinsfile is pushed, we can run the Pipeline as given below:

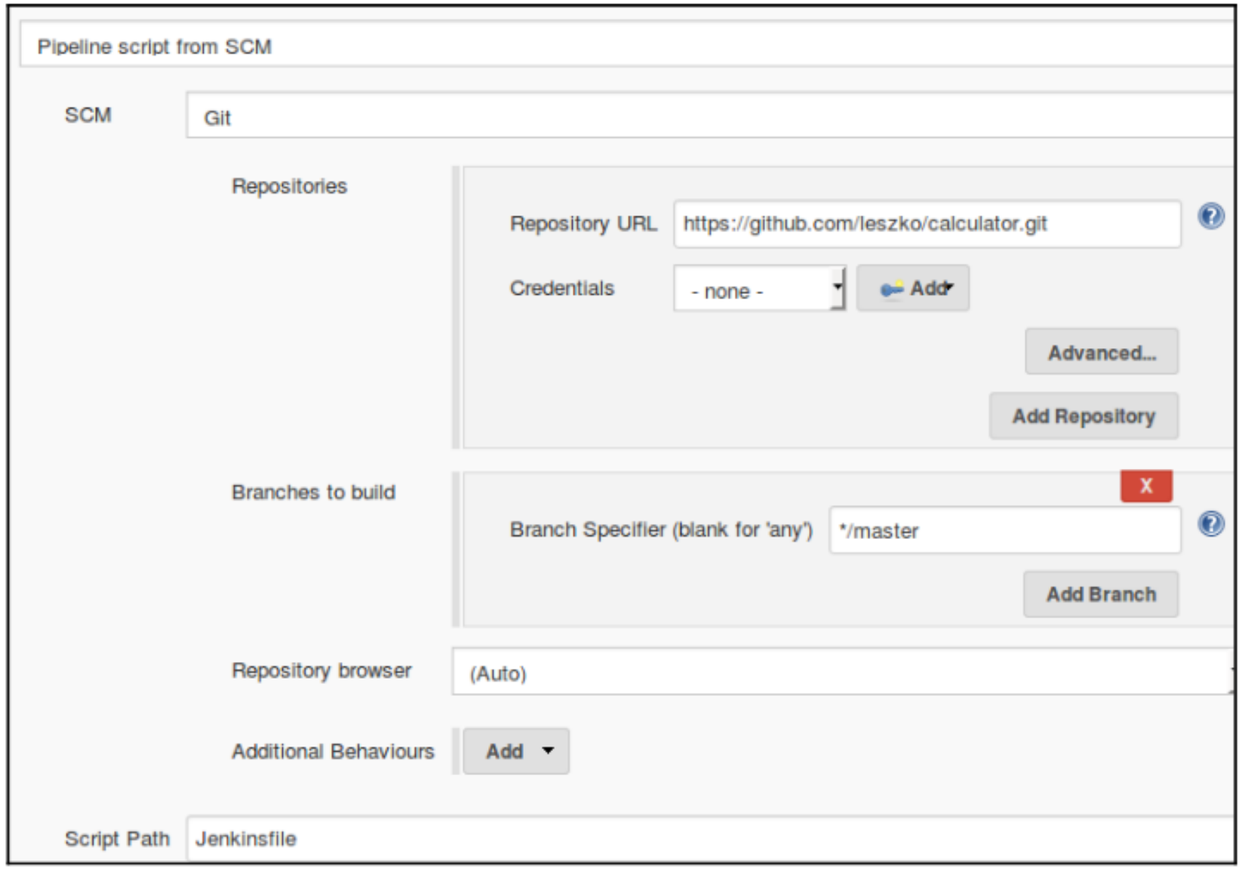
- Open the Pipeline configuration.
- Change *Definition* from *Pipeline script* to *Pipeline script from SCM*.
- Select *Git* in SCM.
- Enter Repository URL.
- Save the Pipeline.

After saving, the build will always run from the current version of Jenkinsfile in the repository.

©Simplilearn. All rights reserved.

Running Jenkinsfile

The screenshot below shows how to run a Pipeline from a Jenkinsfile:



©Simplilearn. All rights reserved.

Build with Jenkinsfile

- The Build stage of the Pipeline will be where source code is assembled, compiled, or packaged.
- The Jenkinsfile is not a replacement for an existing build tool such as GNU/Make, Maven, Gradle, etc.
- It can be considered a glue layer to bind the multiple phases of a project's development lifecycle like build, test, and deploy together.
- Jenkins has plugins for invoking practically any build tool.

©Simplilearn. All rights reserved.

Test with Jenkinsfile

- Running automated tests is a crucial component of any successful continuous delivery process.
- Jenkins has a number of test recording, reporting, and visualization facilities provided by a number of plugins.
- The example below uses the JUnit step, provided by the JUnit plugin. If tests fail, the Pipeline is marked ***unstable***.

```
Jenkinsfile (Declarative Pipeline)
pipeline {
    agent any

    stages {
        stage('Test') {
            steps {
                /* `make check` returns non-zero on test failures,
                * using `true` to allow the Pipeline to continue nonetheless
                */
                sh 'make check || true'
                junit '**/target/*.xml'
            }
        }
    }
}
```

©Simplilearn. All rights reserved.

Test with Jenkinsfile

- Deployment can be anything from publishing built artifacts to an Artifactory server, to pushing code to a production system.
- The example below shows a Deploy Pipeline. At this stage of the example Pipeline, both the *Build* and *Test* stages have been successfully executed.

```
Jenkinsfile (Declarative Pipeline)
pipeline {
        agent any

        stages {
        stage('Deploy') {
        when {
        expression {
        currentBuild.result == null || currentBuild.result == 'SUCCESS'
        }
        }
        steps {
        sh 'make publish'
        }
        }
        }
}
```

©Simplilearn. All rights reserved.

Key Takeaways

You are now able to:

- Describe the importance of continuous integration and continuous deployment
- List the features of Jenkins and demonstrate their uses
- List the features of TeamCity and demonstrate their uses
- Select a suitable build tool for your organization

©Simplilearn. All rights reserved.

Knowledge Check

©Simplilearn. All rights reserved.

Knowledge Check 1	What advantage does Maven have over Ant?

a. There isn't one

b. Ant only compiles code

c. Maven is easier to configure

d. It resolves dependencies

©Simplilearn. All rights reserved.

Knowledge Check 1	What advantage does Maven have over Ant?

a. There isn't one

b. Ant only compiles code

c. Maven is easier to configure

d. It resolves dependencies

The correct answer is **d. It resolves dependencies**

It resolves dependencies on external jar files.

©Simplilearn. All rights reserved.

Knowledge Check 2	What advantage does continuous integration provide?

a. It simplifies the build process

b. It stops developers checking in bad code

c. Build errors are quickly detected and reported

d. There are no real advantages

©Simplilearn. All rights reserved.

Knowledge Check 2	What advantage does continuous integration provide?

a. It simplifies the build process

b. It stops developers checking in bad code

c. Build errors are quickly detected and reported

d. There are no real advantages

The correct answer is **c. Build errors are quickly detected and reported**

Continuous integration can also perform testing and can generate documentation.

©Simplilearn. All rights reserved.

Knowledge Check 3	What does POM stand for?

a. Project Object Model

b. Project Oriented Model

c. Project Operational Model

d. Purpose Only Manufacturing

©Simplilearn. All rights reserved.

Knowledge Check 3	What does POM stand for?

a. Project Object Model

b. Project Oriented Model

c. Project Operational Model

d. Purpose Only Manufacturing

The correct answer is **a. Project Object Model**

©Simplilearn. All rights reserved.

Knowledge Check 4	What is continuous deployment?

a. A deployment server

b. Deployment tool

c. Open source deployment server for containers

d. Minimizing the time elapsed between writing new code and using new code in production

©Simplilearn. All rights reserved.

Knowledge Check 4	What is continuous deployment?

a. A deployment server

b. Deployment tool

c. Open source deployment server for containers

d. Minimizing the time elapsed between writing new code and using new code in production

The correct answer is **d. Minimizing the time elapsed between writing new code and using new code in production**

Continuous deployment refers to automated, faster, and quicker deployments of code into production.

©Simplilearn. All rights reserved.

Knowledge Check 5	Which of the following is NOT a continuous deployment tool?

a. Microsoft Visual Studio

b. GitHub

c. ElectricFlow

d. Bamboo

©Simplilearn. All rights reserved.

Knowledge Check 5	Which of the following is NOT a continuous deployment tool?

a. Microsoft Visual Studio

b. GitHub

c. ElectricFlow

d. Bamboo

The correct answer is **b. GitHub**

GitHub is not a continuous deployment tool as it is a source code repository.

©Simplilearn. All rights reserved.

Lesson-End Project

Continuous Integration with Jenkins, Git, and Maven

Duration: 60 mins

Problem Statement:

Create a FreeStyle project in Jenkins and complete the following:

- Install “Email Extension plugin” in Jenkins.
- Configure Gmail in Jenkins.
- Receive an email when build fails and succeeds.

Access: Click on the **Labs** tab on the left side panel of the LMS. Copy or note the username and password that is generated. Click on the **Launch Lab** button. On the page that appears, enter the username and password in the respective fields, and click **Login**.

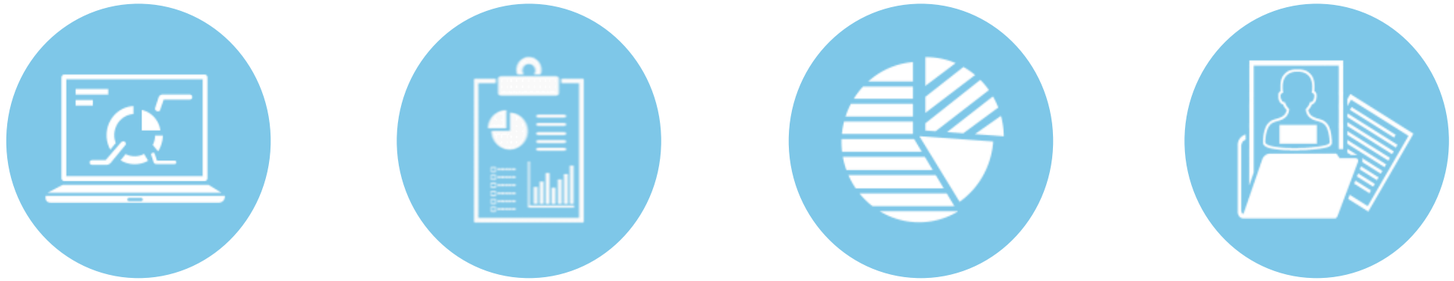
©Simplilearn. All rights reserved.

Thank You

©Simplilearn. All rights reserved.

DevOps Certification Training

Lesson 04: Software and Automation Testing Frameworks



© Simplilearn. All rights reserved.

Learning Objectives

By the end of this lesson, you will be able to:

- Describe the benefits in traditional and agile approach of software testing
- Describe the levels and approaches of software testing
- Demonstrate test-driven development framework with JUnit
- Demonstrate behavior-driven development framework with cucumber

©Simplilearn. All rights reserved.

Software and Automation Testing Frameworks

Software Testing Overview

©Simplilearn. All rights reserved.

simplilearn

Software Testing

Software testing is an investigation conducted to share information about the quality of the product and the service under test to the organization.

Is it meeting the requirements that guided its design and development?

Is it responding correctly to the inputs?

Is it functioning within an acceptable time?

Is it working properly in its intended environments?

Is it achieving the results?

©Simplilearn. All rights reserved.

Source:Wikipedia

simplilearn

Traditional vs. Agile Testing

Traditional Testing

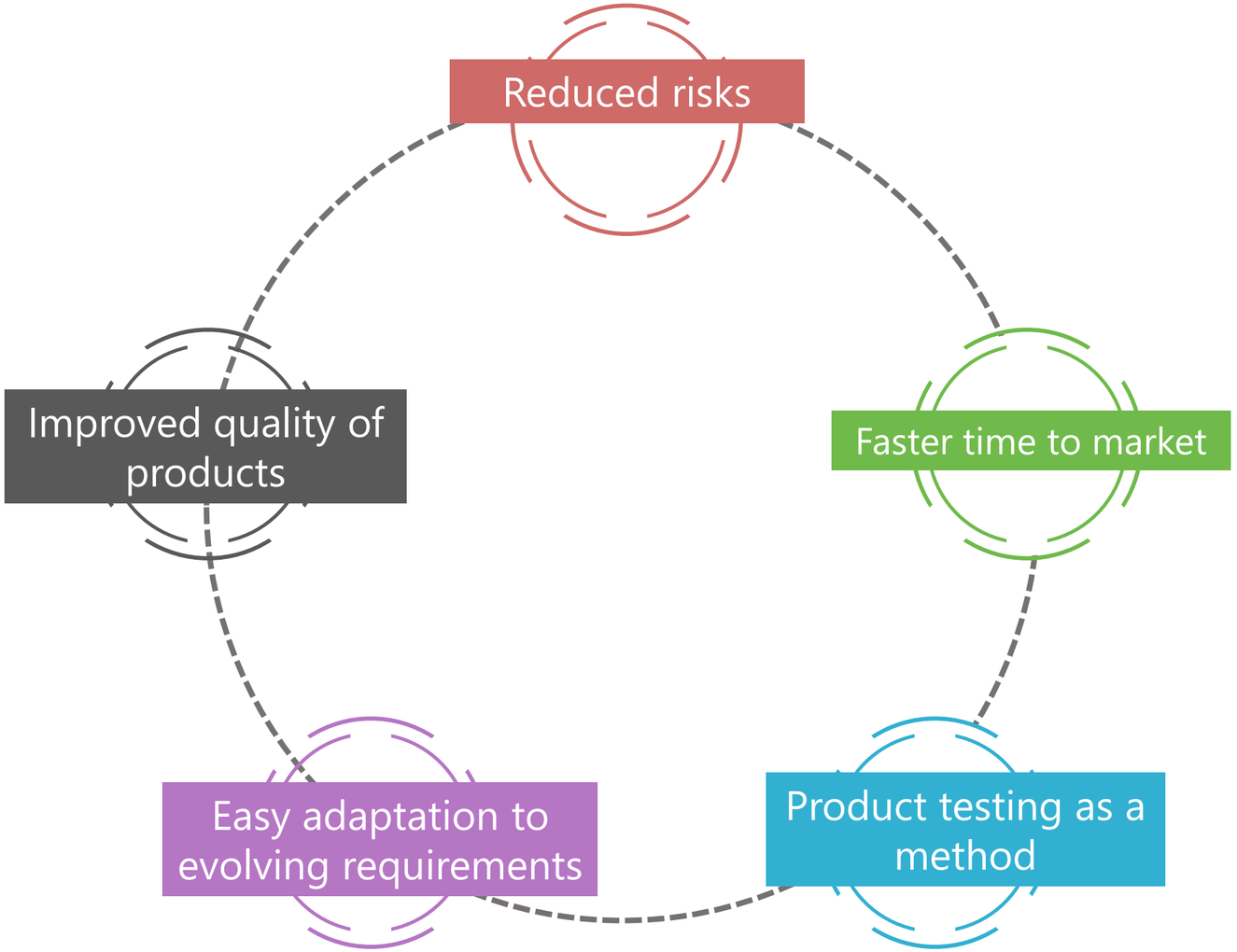
- Follows predictive model with a step-by-step phased approach
- Certifies the quality of product
- Produces the final system with all the required features
- Unit testing for each module followed by integration and system testing
- No feedback until test is completed
- Modifications are implemented in the next release

Agile Testing

- Follows adaptive model with iterative approach
- Certifies the quality and fast delivery of the product with minimal functionalities
- Produces working versions of the final system that have a subset of the features
- Continuous testing and regression across all iterations
- At the end of every sprint, short ongoing feedback cycles are produced
- Modifications are implemented in the next sprint of the test cycle

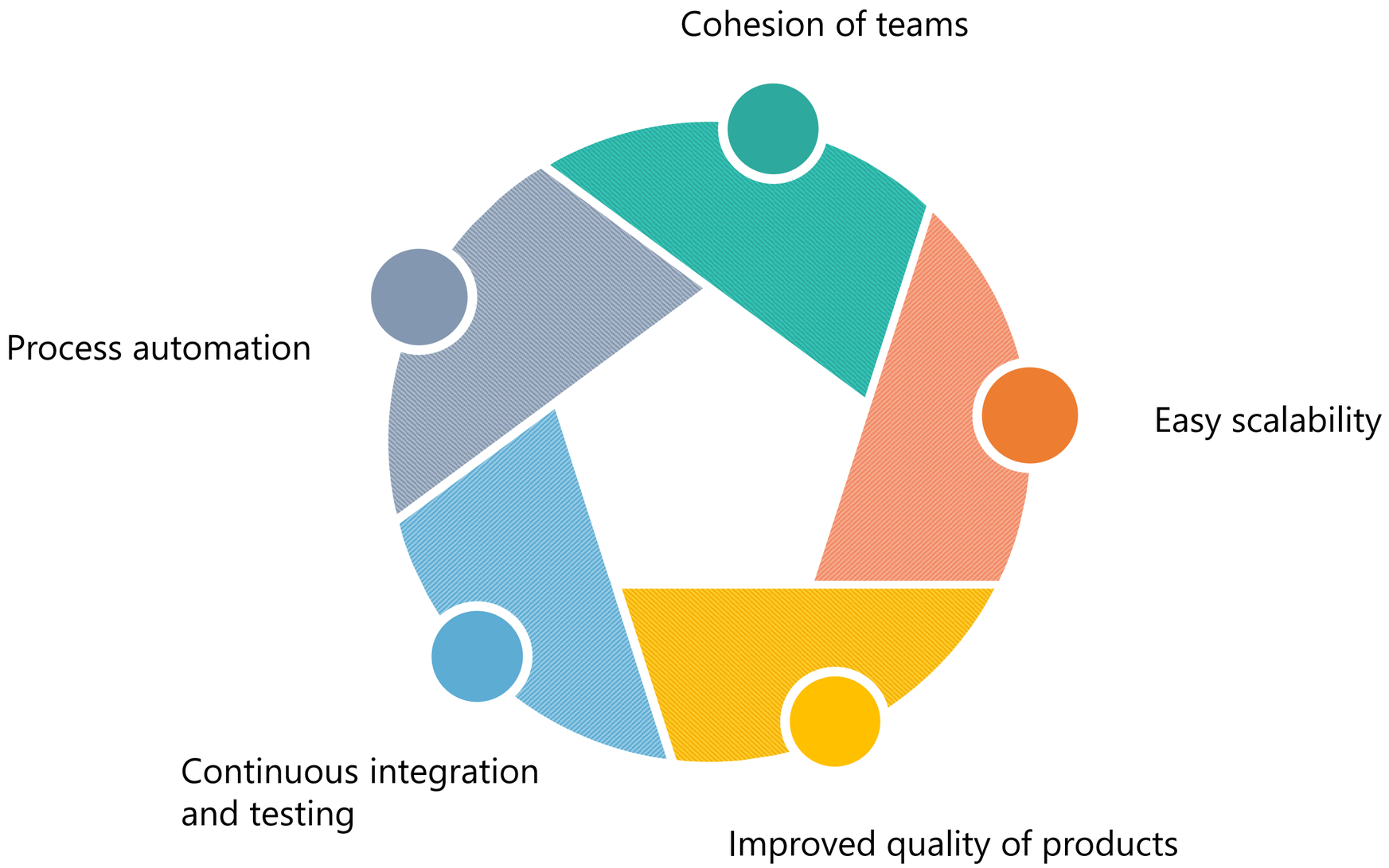
©Simplilearn. All rights reserved.

Agile Testing Advantages



©Simplilearn. All rights reserved.

Advantages of Using Agile with DevOps Testing



©Simplilearn. All rights reserved.

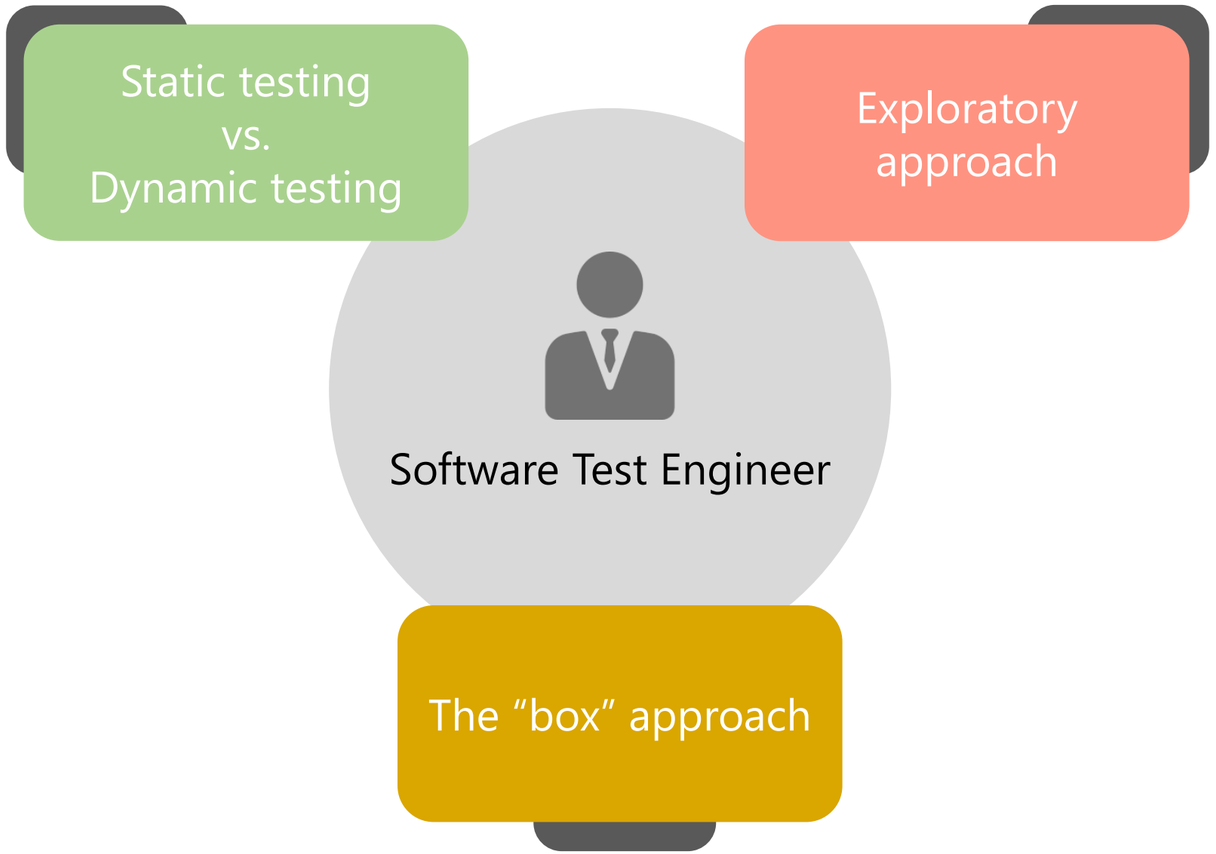
Software and Automation Testing Frameworks

Testing Levels, Approaches, and Automation Tools

©Simplilearn. All rights reserved.

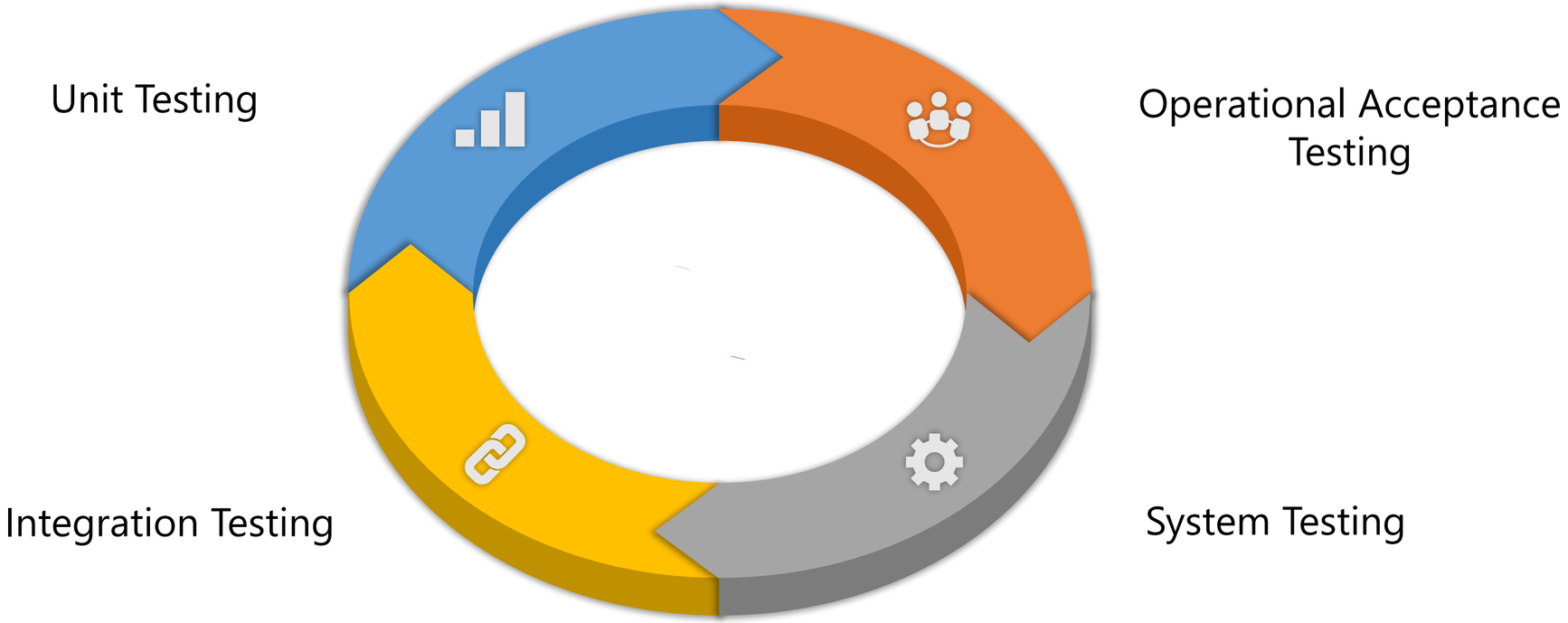
simplilearn

Software Testing Approaches



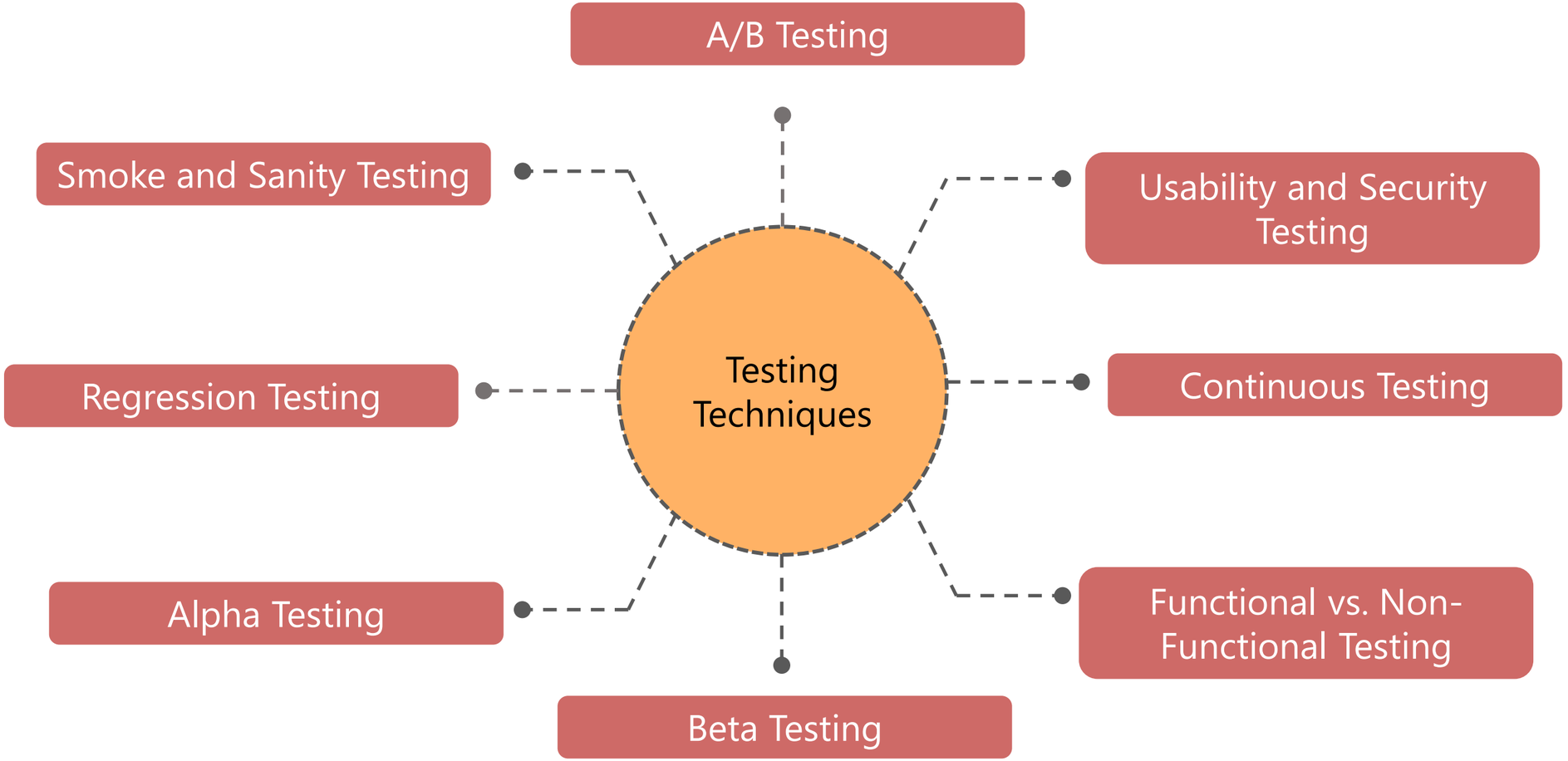
©Simplilearn. All rights reserved.

Software Testing Levels



©Simplilearn. All rights reserved.

Popular Software Testing Techniques



©Simplilearn. All rights reserved.

Automation Testing Framework

Test-driven development teams rely on automated testing. A testing framework is used to write test cases and a continuous integration software will run these test cases automatically. A well-developed testing framework or an integrated development environment (IDE) can be used to run the test scripts.

TestNG

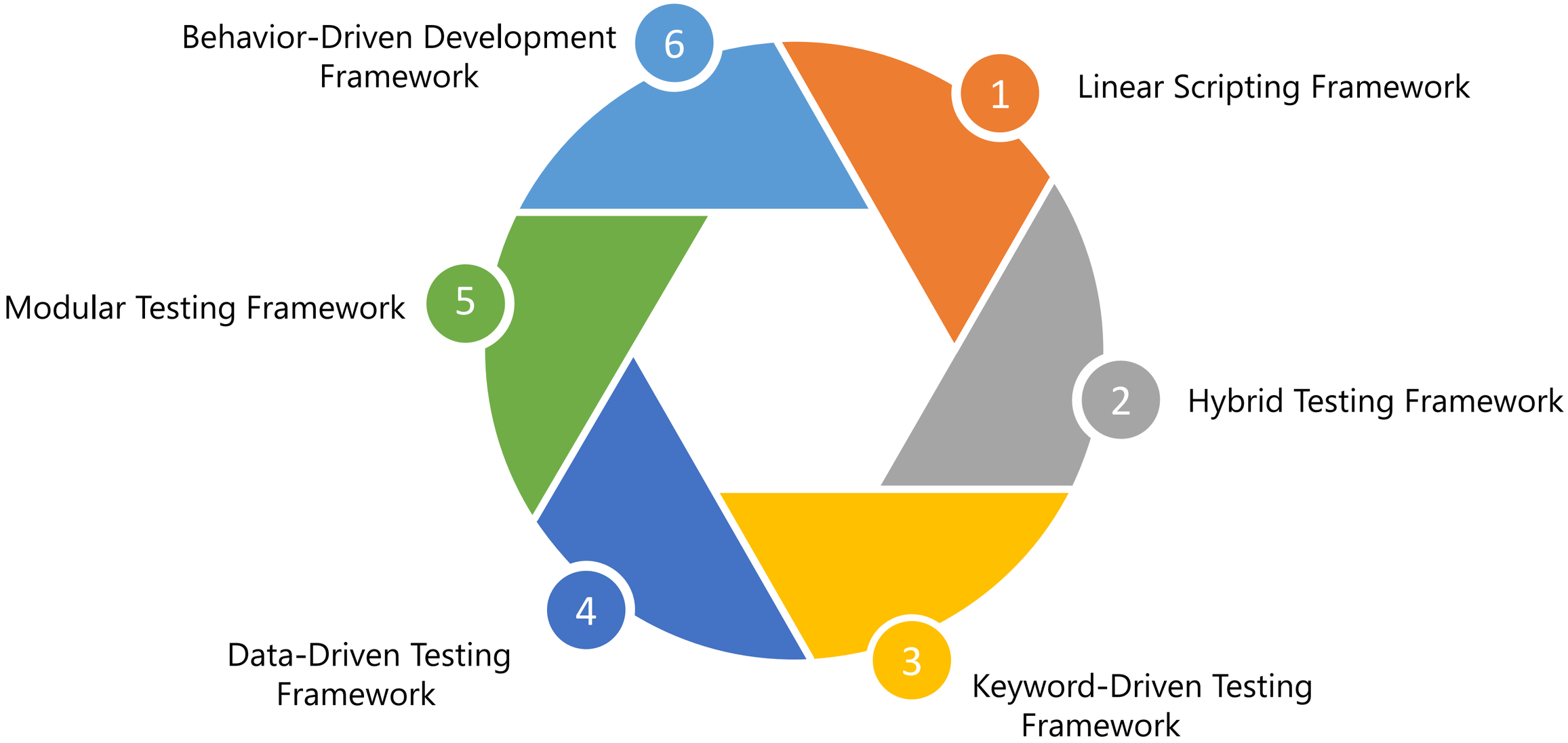
appium

Jenkins

JUnit 5

©Simplilearn. All rights reserved.

Automation Testing Framework (Contd.)



©Simplilearn. All rights reserved.

Popular Testing Tools

SoapUI

SoapUI is an open-source web service testing application for service-oriented architectures and RESTful APIs.

Robot Framework is a keyword-driven test automation framework that uses tabular test data syntax.

©Simplilearn. All rights reserved.

Popular Testing Tools (Contd.)

RSpec is a domain specific testing tool to test Ruby code. It is a behavior-driven development framework that is used in the production applications.

Eggplant Functional is a black-box GUI test automation tool which uses intelligent image recognition algorithms to monitor the display being tested. It is used for mobile testing, cross-platform testing, and performance testing.

©Simplilearn. All rights reserved.

Popular Testing Tools (Contd.)

Mocha is a JavaScript test framework for Node.js programs. Its features are browser support, asynchronous testing, test coverage reports, and use of any assertion library.

Jasmine is an open-source testing framework for JavaScript. It supports asynchronous testing, use of 'spies' for implementing test doubles, testing of front-end code through a front-end extension.

©Simplilearn. All rights reserved.

Software and Automation Testing Frameworks

Test-Driven Development Approaches and JUnit 5

©Simplilearn. All rights reserved.

simplilearn

Overview of Test-Driven Development

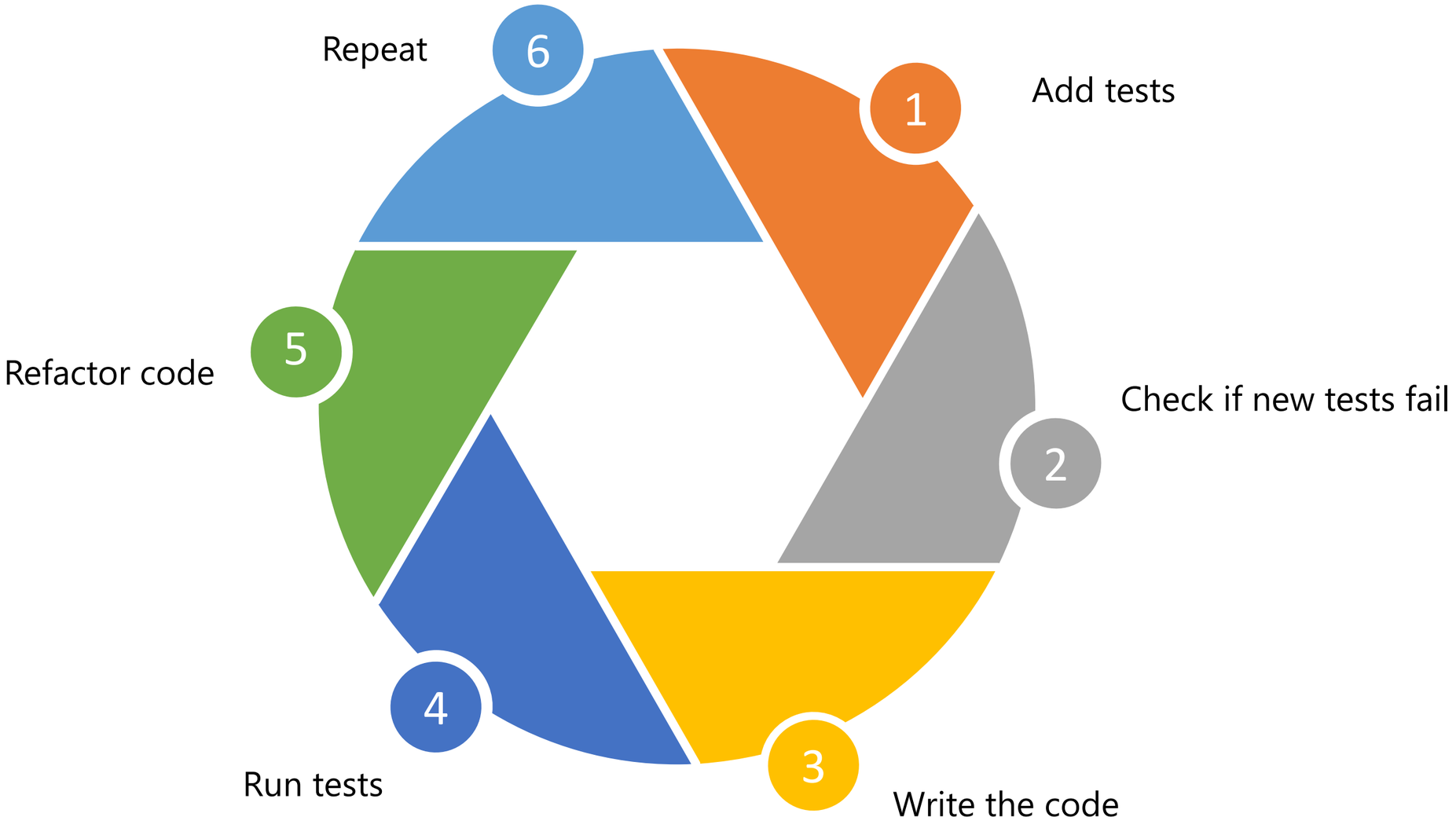
Test-driven development is a software development process that depends on the repetition of short development cycle. Requirements are converted into test cases, then the software is improved to pass new tests. Developers apply the concept to improve and debug legacy code developed in older techniques.

©Simplilearn. All rights reserved.

Source:Wikipedia

simplilearn

Test-Driven Development Cycle



©Simplilearn. All rights reserved.

Test-Driven Development Advantages

- Fast Feedback
- Creation of detailed specification
- Less rework
- Less debugging
- Fast identification of errors
- Creation of solid code
- Clear Interface
- Maintainable and Flexible
- Faster Time to market
- Less Development cost
- Improved Quality
- Increase in Programmer's productivity
- Less bugs
- Simplification of codes
- Adaptability

©Simplilearn. All rights reserved.

General Approach

Test-driven development is related to the test-first programming concept of extreme programming. Recently, it has created general interest and is applied with minor semantic changes in the approach.

1. Add a check replaces add a test
2. Run all checks replaces run all tests
3. Do the work replaces write some code
4. Run all checks replaces run tests
5. Clean up the work replaces refactor code
6. Repeat

©Simplilearn. All rights reserved.

Overview of JUnit

Junit is a unit testing framework for the Java programming language. It is found that 10,000 Java projects used JUnit as an external library. Each library of the JUnit was used by 30.7% of the projects. It is the first option for developers while testing the Java applications.

JUnit 5

©Simplilearn. All rights reserved.

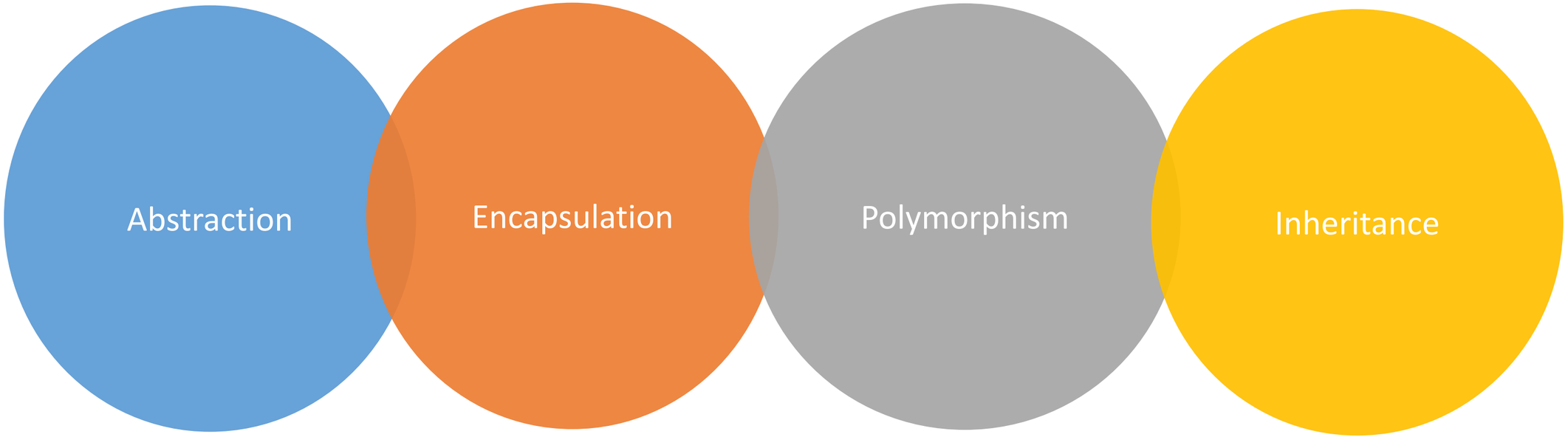
Advantages of JUnit

- The server requirement is eliminated to test web applications
- Hierarchy of the program code can be tested as a single unit or multiple units
- Supports test assertions and immediate test reporting
- Simple framework for writing automated tests
- Availability of a text-based command line and graphical test reporting mechanism

©Simplilearn. All rights reserved.

Java: A Programming Language

Java is an object-oriented programming language. It is intended to let the developers “write once, run anywhere”. Java applications are compiled to bytecode that can run on any Java virtual machine (JVM).



©Simplilearn. All rights reserved.

Java Program Template

A Java program is a collection of objects, classes, methods, and instance variables.

Syntax of Java program

```
Packages;
Access_Modifiers keyword Classname {

    Global variables;
    Access_modifiers keyword datatype functionName(Argument(s)){

        //statements and expressions;

    }
}
```

©Simplilearn. All rights reserved.

Junit 5 with Java

JUnit 5 + Java

Software required:

- Java SE (Java 8 or higher versions)
- Eclipse IDE
- JUnit library

©Simplilearn. All rights reserved.

Assisted Practice

Duration: 30 mins

Test-Driven Development Approach with JUnit 5

Problem Statement: You are given a project to demonstrate the test-driven development approach using JUnit 5 and Java as a programming language.

Access: Click on the **Labs** tab on the left side panel of the LMS. Copy or note the username and password that is generated. Click on the **Launch Lab** button. On the page that appears, enter the username and password in the respective fields, and click **Login**.

©Simplilearn. All rights reserved.

Assisted Practice: Guidelines to Demonstrate the Usage of Junit 5 with Java

1. Download Java 8+ version and set up the environment variables.
2. Download eclipse from the official eclipse site.
3. Create a new Java project and refer the guide document, 4.1-JUnit demonstration with Java.
4. Create 2 classes to add and subtract numbers.
5. Generate the JUnit test files for each classes (Add the files in a different source directory for better indentation).
6. Confirm eclipse to add JUnit 5 library to the build path.
7. Confirm that the test cases are failed and the error logs are available in “Failure Trace” section in eclipse.
8. In the TestSrc folder, remove the comments (3 and 5) and uncheck the comments (1, 2, 4, 6).
9. Re-run the test cases and confirm that the tests are successful without any error or warning logs in the Failure Trace.

©Simplilearn. All rights reserved.

Software and Automation Testing Frameworks

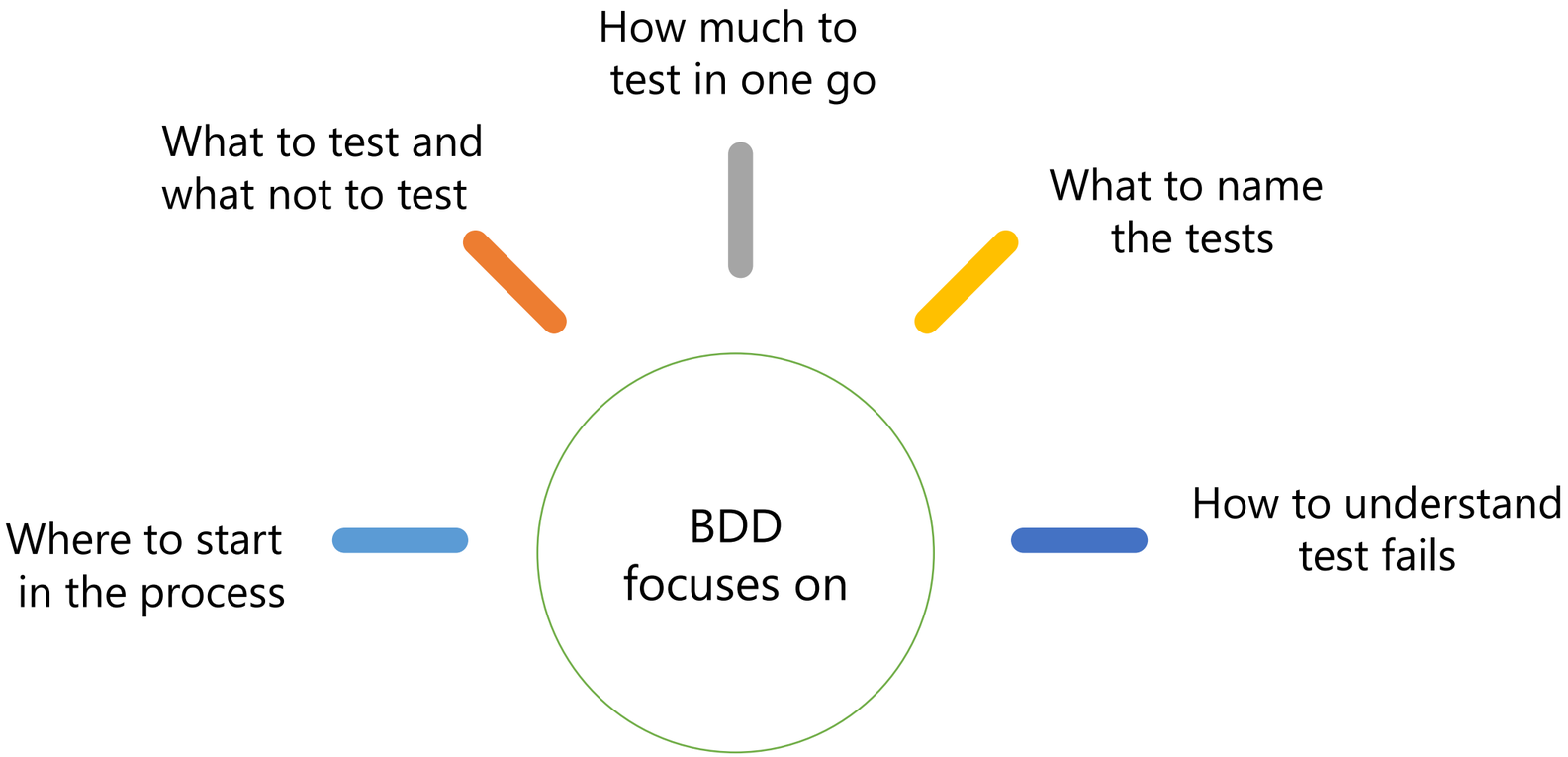
Behavior-Driven Development Principles, Cucumber, and its Applications

©Simplilearn. All rights reserved.

simplilearn

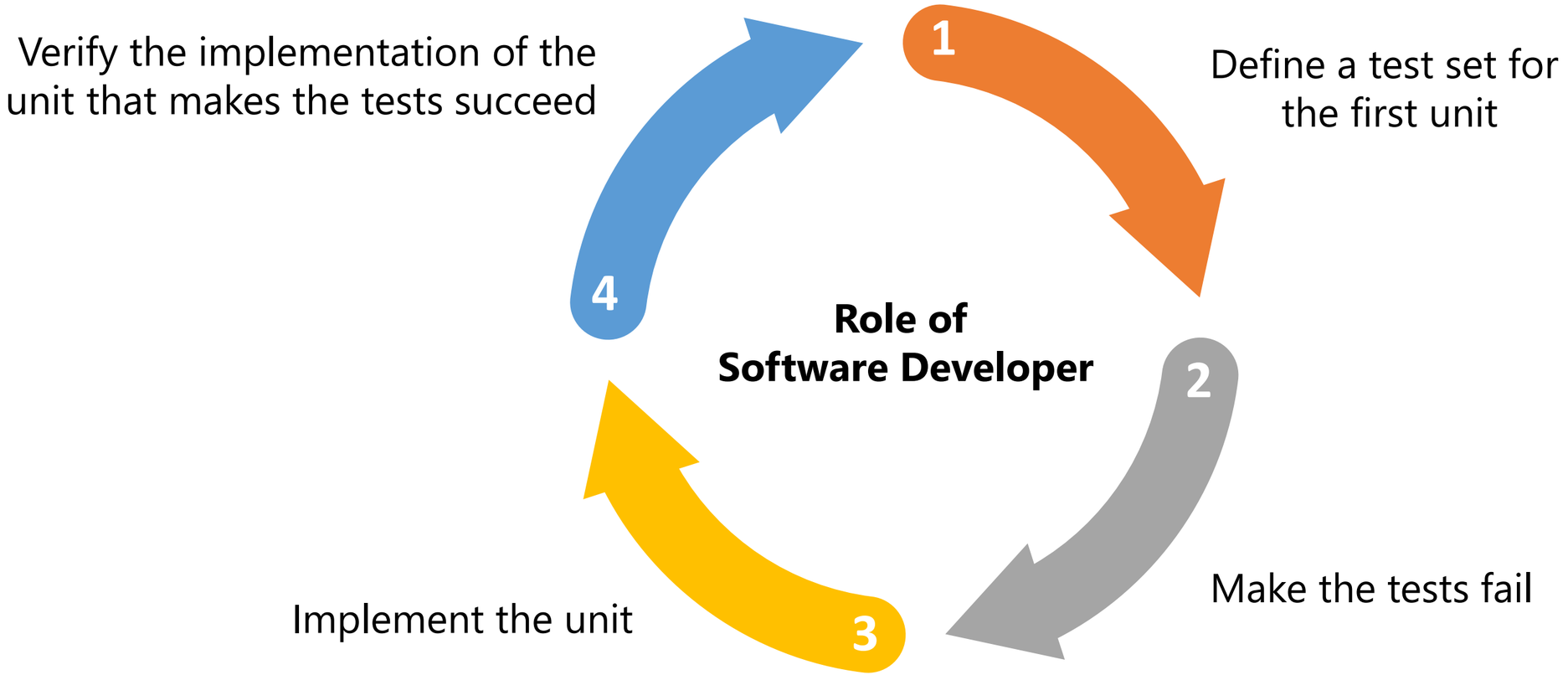
Overview of Behavior-Driven Development

Behavior-driven development is an extension of test-driven development.



©Simplilearn. All rights reserved.

Behavior Driven-Development Principles



©Simplilearn. All rights reserved.

Overview of Cucumber

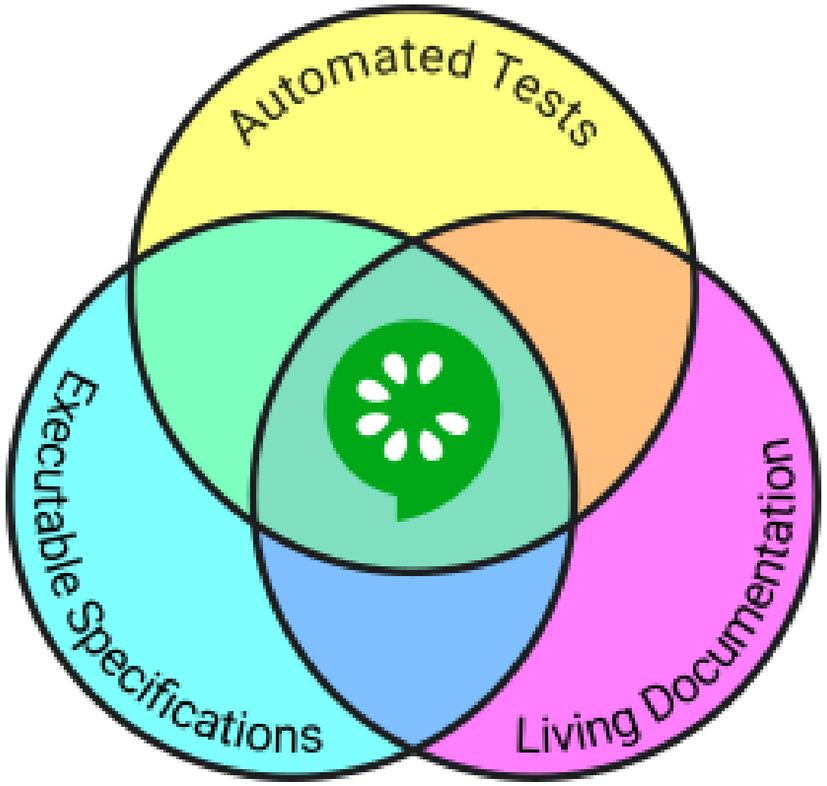
Cucumber is a software tool for testing other software by running automated acceptance tests written in a behavior-driven development approach. It allows the execution of feature documentation written in business-facing text. Gherkin language is used to define test cases. It is designed for non-technical, human readable, and collectively describes use cases related to software system.

cucumber

©Simplilearn. All rights reserved.

Gherkin: A Language for Cucumber

Gherkin is a set of grammar rules that structures the plain text for cucumber to understand. It serves multiple purposes: Unambiguous executable specification, automated testing, and documentation about the system's actual behavior.



©Simplilearn. All rights reserved.

Gherkin Keywords

Each line of the script must start with a Gherkin keyword followed by any relevant text.

The primary keywords are:

- Feature
- Scenario
- Given, When, Then, And, But (Steps)
- Background
- Scenario Outline
- Examples

The secondary keywords are:

- “ “ “ (Doc Strings)
- | (Data Tables)
- @ (Tags)
- # (Comments)

©Simplilearn. All rights reserved.

Tools for Cucumber

Supported Editors

Supported IDEs

Supported Build Tools

©Simplilearn. All rights reserved.

Cucumber with Java

Software required:

- Java SE (Java 9 and higher versions are not supported)
- Maven – Version 3.3.1 or higher
- IntelliJ IDEA Cucumber for Java plugin
- Cucumber Eclipse (An alternative for IntelliJ)

©Simplilearn. All rights reserved.

Cucumber with JavaScript

Software required:

- Node.js – A JavaScript runtime engine
- Text Editor – Atom, Visual Studio Code, or TextMate (For Mac Users Only)

©Simplilearn. All rights reserved.

Cucumber with Ruby

Software required:

- Ruby
- Bundler
- Text Editor

©Simplilearn. All rights reserved.

Cucumber with Kotlin

Software required:

- Java SE (Java 9 and higher versions are not supported)
- Maven – Version 3.3.1 or higher
- IntelliJ IDEA Cucumber for Java plugin and IntelliJ IDEA Kotlin plugin
- Cucumber Eclipse and Kotlin Eclipse (An alternative for IntelliJ)

©Simplilearn. All rights reserved.

Assisted Practice

Duration: 45 mins

Behavior-Driven Development Approach with Cucumber

Problem Statement: You are given a project to demonstrate the behavior-driven development approach using cucumber and JavaScript as a language.

Access: Click on the **Labs** tab on the left side panel of the LMS. Copy or note the username and password that is generated. Click on the **Launch Lab** button. On the page that appears, enter the username and password in the respective fields, and click **Login**.

©Simplilearn. All rights reserved.

Assisted Practice: Guidelines to Demonstrate the Usage of Cucumber

1. Install node.js and npm from the node.js official website.
2. Open the Visual Studio Code terminal and check the version of node and npm.
3. Create an empty cucumber project folder.
4. Add cucumber as a development dependency.
5. Create a folder “feature” and a sub-folder “steps” in it.
6. Add the codes in the respective files. (Codes are available in the guide Document, 4.2 – Cucumber installation).
7. Verify cucumber installation.
8. Follow the guide document, 4.3 – Cucumber application with JavaScript
9. Confirm that all the scenarios and steps are passed.

©Simplilearn. All rights reserved.

Key Takeaways

You are now able to:

- Describe the benefits in traditional and agile approach of software testing
- Describe the levels and approaches of software testing
- Demonstrate test-driven development framework with JUnit
- Demonstrate behavior-driven development framework with cucumber

©Simplilearn. All rights reserved.

Knowledge Check

©Simplilearn. All rights reserved.

Knowledge Check 1	_____ **set of JUnit tests focuses on test-driven development.**

a. Grey-box

b. Sanity

c. Structure-based

d. All of the above

©Simplilearn. All rights reserved.

Knowledge Check 1	_____ **set of JUnit tests focuses on test-driven development.**

a. Grey-box

b. Sanity

c. Structure-based

d. All of the above

The correct answer is **c. Structure-based**

Structure-based testing, tests the structure of the code and is a set of JUnit tests focused on test-driven development.

©Simplilearn. All rights reserved.

Knowledge Check 2	**White-box testing is carried out to test the _____ components.**

a. Dynamic

b. Static

c. Final

d. All the above

©Simplilearn. All rights reserved.

Knowledge Check 2	**White-box testing is carried out to test the _____ components.**

a. Dynamic

b. Static

c. Final

d. All the above

The correct answer is **a. Dynamic**

White-box testing, tests the dynamic components and the internal structure of the code.

©Simplilearn. All rights reserved.

Knowledge Check 3	**Which of the following is triggered by migrating the existing software?**

a. Maintenance Testing

b. Sanity Testing

c. White-box Testing

d. None of the above

©Simplilearn. All rights reserved.

Knowledge Check

3

Which of the following is triggered by migrating the existing software?

a. Maintenance Testing

b. Sanity Testing

c. White-box Testing

d. None of the above

The correct answer is **a. Maintenance Testing**

Maintenance testing is in charge of the testing on the already deployed software.

©Simplilearn. All rights reserved.

Knowledge Check 4	**Cucumber software is written in _____ form.**

a. C

b. Java

c. C++

d. Ruby

©Simplilearn. All rights reserved.

Knowledge Check 4	**Cucumber software is written in _____ form.**

a. C

b. Java

c. C++

d. Ruby

The correct answer is **d. Ruby**

©Simplilearn. All rights reserved.

Knowledge Check 5	**What is feature file in Cucumber?**

a. Feature

b. Scenario

c. Scenario Outline

d. All of the above

©Simplilearn. All rights reserved.

Knowledge Check

5

What is feature file in Cucumber?

a. Feature

b. Scenario

c. Scenario Outline

d. All of the above

The correct answer is **d. All of the above**

Feature file is an entry point to all the cucumber tests. It is used to describe the tests in Gherkin language.

©Simplilearn. All rights reserved.

Lesson-End Project

Behavior-Driven Development Approach with Java and Kotlin

Duration: 90 mins

Problem Statement:

Create a directory named **Lesson-04** and perform the following:

- Install cucumber on IntelliJ for Java and Visual Studio Code for Kotlin.
- Execute the 3 scenarios: Undefined, Failed, and Passed for Java.
- Execute the 3 scenarios: Undefined, Failed, and Passed for Kotlin.

Access: Click on the **Labs** tab on the left side panel of the LMS. Copy or note the username and password that is generated. Click on the **Launch Lab** button. On the page that appears, enter the username and password in the respective fields, and click **Login**.

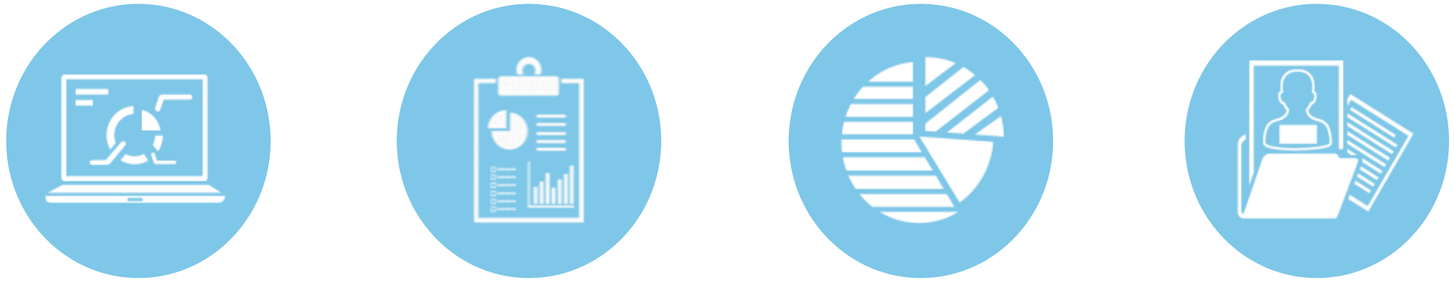
©Simplilearn. All rights reserved.

Thank You

©Simplilearn. All rights reserved.

DevOps Certification Training

Lesson 05: Configuration Management Tools



© Simplilearn. All rights reserved.

Learning Objectives

By the end of this lesson, you will be able to:

- Explain the concepts of configuration management tools
- Demonstrate Puppet installation
- Demonstrate Chef installation
- Demonstrate Ansible installation and its usage
- Describe SaltStack
- Select the suitable CM tool for your organization

©Simplilearn. All rights reserved.

Configuration Management Tools

Overview of Configuration Management Tools

©Simplilearn. All rights reserved.

Configuration Management Tools

- Configuration management tools manage all configuration items in a software for all environments.
- These configuration items can be software application files, software packages, and software installations which need to be configured for specific environments.
- Configuration management tools cover both, software and server configurations.
- They help enforce standardization in software configurations without any errors.
- They also help to reduce the time taken to manage configurations manually on each and every server.

©Simplilearn. All rights reserved.

Popular Configuration Management Tools

CHEF

puppet

SALTSTACK

ANSIBLE

©Simplilearn. All rights reserved.

Purpose of Configuration Management Tool

1. Helps to manage software configurations effectively
2. Helps to remove manual errors while performing configuration changes for application software
3. Helps to manage configurations in multiple environments
4. Generic configurations can be stored in version control so that they can be used in other projects as well
5. Saves time by automating generic configuration procedures
6. CM tools can be used to deploy application source codes across vast infrastructure
7. Eliminates the need of documentation process for performing configuration changes

©Simplilearn. All rights reserved.

Configuration Management Process

Configuration Regulation
Regulates the way configuration changes are made to application software

Configuration Identification
Identifies the correct configuration that needs to be managed by CM tool

Configuration Compliance
Audits and implements compliance on configuration changes made to application software

©Simplilearn. All rights reserved.

Configuration Management Tools

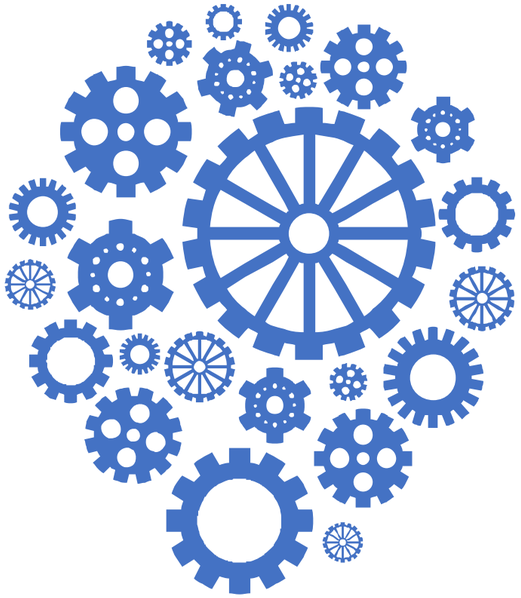
Managing Infrastructure

©Simplilearn. All rights reserved.

Role of Infrastructure as Code in DevOps Environment

Modern way to manage configuration items

Documentation for software modifications and infrastructure configurations



Admins can manage multiple environments with infrastructure scripts/codes

Considered an essential component of DevOps

Easy to integrate with version control and share with others

©Simplilearn. All rights reserved.

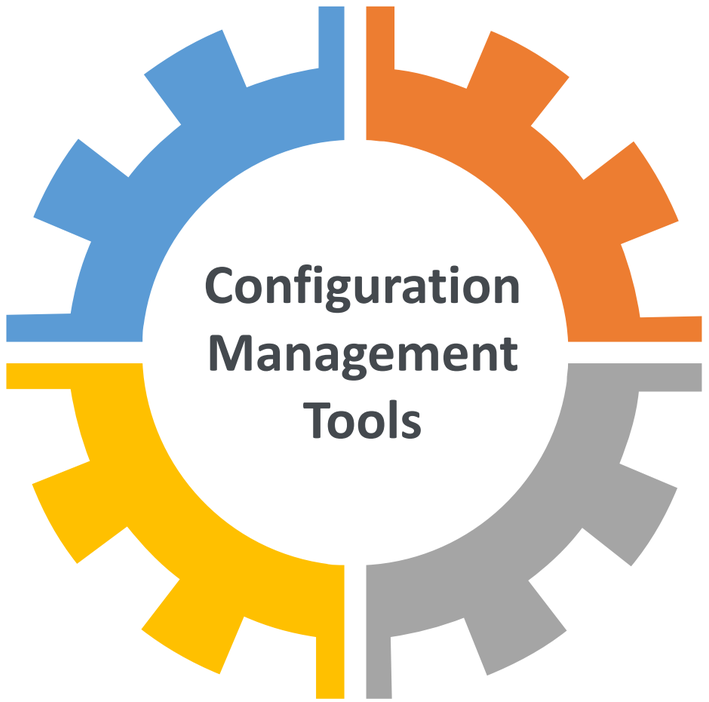
Configuration Management Tools

Types of Configuration Management tools

©Simplilearn. All rights reserved.

Types of Configuration Management Tools

Puppet
Ruby DSL-based CM tool used for managing software, systems, and network configuration items



Chef
Ruby-based CM tool having integration with most of cloud-based platforms

Ansible
Python-based CM tool, also considered as agentless CM tool

SaltStack
Python-based open-source CM tool used to manage configuration items remotely

©Simplilearn. All rights reserved.

Configuration Management Tools

Overview of Puppet

©Simplilearn. All rights reserved.

Puppet

Puppet is an open-source CM Tool

Developed and launched by Puppet Labs in year 2005

Written in Ruby DSL language. It is launched under GNU and Apache license 2.0

Used to manage both, Unix and Windows-based servers

Follows client-server architecture for managing configuration items

Designed to manage changes on software systems

Used to manage various resources and application software installations

©Simplilearn. All rights reserved.

Manifest File in Puppet

Manifest file in puppet is used to manage various resources on software systems. These resources are packages, files, and system configuration changes. Resources can be managed using the manifest script file written in Ruby DSL language.

```
# execute 'apt-get update'
exec { 'apt-update':                    # exec resource named 'apt-update'
  command => '/usr/bin/apt-get update'  # command this resource will run
}

# install apache2 package
package { 'apache2':
  require => Exec['apt-update'],        # require 'apt-update' before installing
  ensure => installed,
}

# ensure apache2 service is running
service { 'apache2':
  ensure => running,
}
```

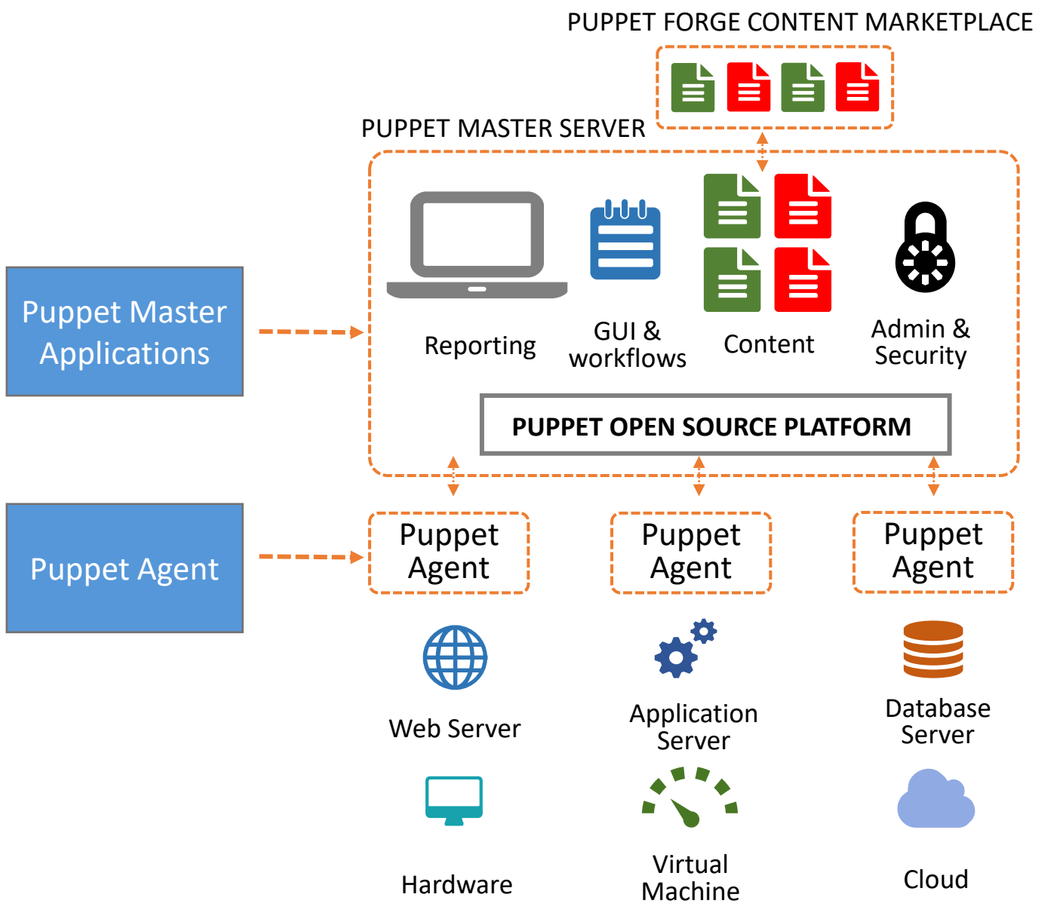
©Simplilearn. All rights reserved.

Manifest File in Puppet (Contd.)

```
# install mysql-server package
package { 'mysql-server':
  require => Exec['apt-update'],        # require 'apt-update' before installing
  ensure => installed,
}

# ensure mysql service is running
service { 'mysql':
  ensure => running,
}
# install php5 package
package { 'php':
  require => Exec['apt-update'],        # require 'apt-update' before installing
  ensure => installed,
}
# ensure info.php file exists
file { '/var/www/html/info.php':
  ensure => file,
  content => '<?php  phpinfo(); ?>',    # phpinfo code
  require => Package['apache2'],        # require 'apache2' package before creating
}
```

©Simplilearn. All rights reserved.

Puppet Architecture



- Describes the desired system state of the computer. Examples: Software, libraries, and configuration setup.
- Lists all resources as well as dependencies between those resources in a catalog. Example: A Windows OS upgrade will affect the software Visual Studio Code.
- Puppet uses several resources of information to compile a catalog.

©Simplilearn. All rights reserved.

Configuration Management Tools

Overview of Chef

©Simplilearn. All rights reserved.

Chef Architecture

- Chef was developed in Ruby and Erlang languages. It was initially launched in year 2009.
- It uses pure-Ruby DSL based language for managing system resources.
- It’s extensive integration with latest cloud platforms like AWS, GoogChef runs both in Client-Server and solo architecture for managing both infra and software resources.
- We can manage both Unix based and Windows based systems using Chef.

©Simplilearn. All rights reserved.

Chef Architecture (Contd.)

- Chef is comprised of three main components:

  Chef-server, Chef-client, and Workstation.

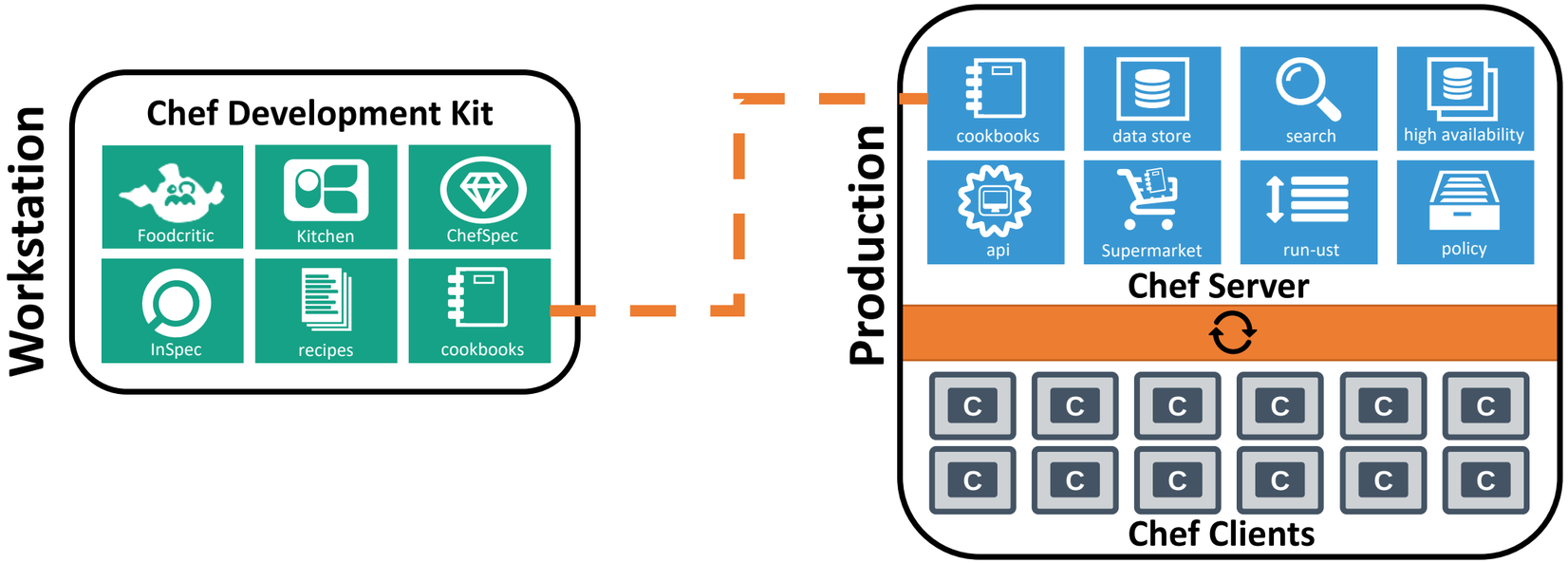
- Chef-server is used to manage various chef resources, information on nodes, and cookbooks.

- Chef-client should be deployed on various systems in a Chef environment.

- Workstation is one of the components used to manage all Chef resources.

©Simplilearn. All rights reserved.

Chef Architecture (Contd.)



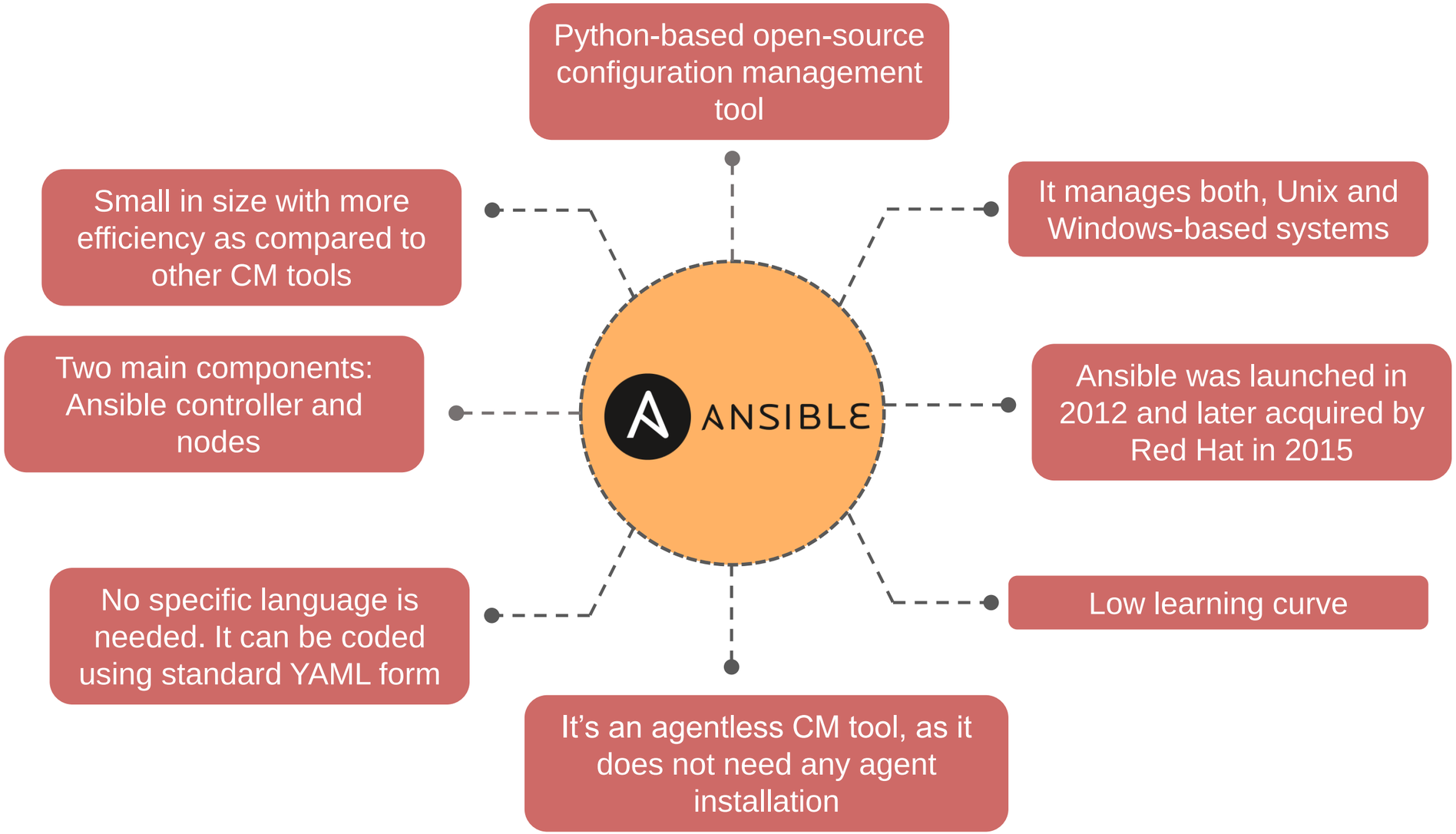
©Simplilearn. All rights reserved.

Configuration Management Tools

Overview of Ansible

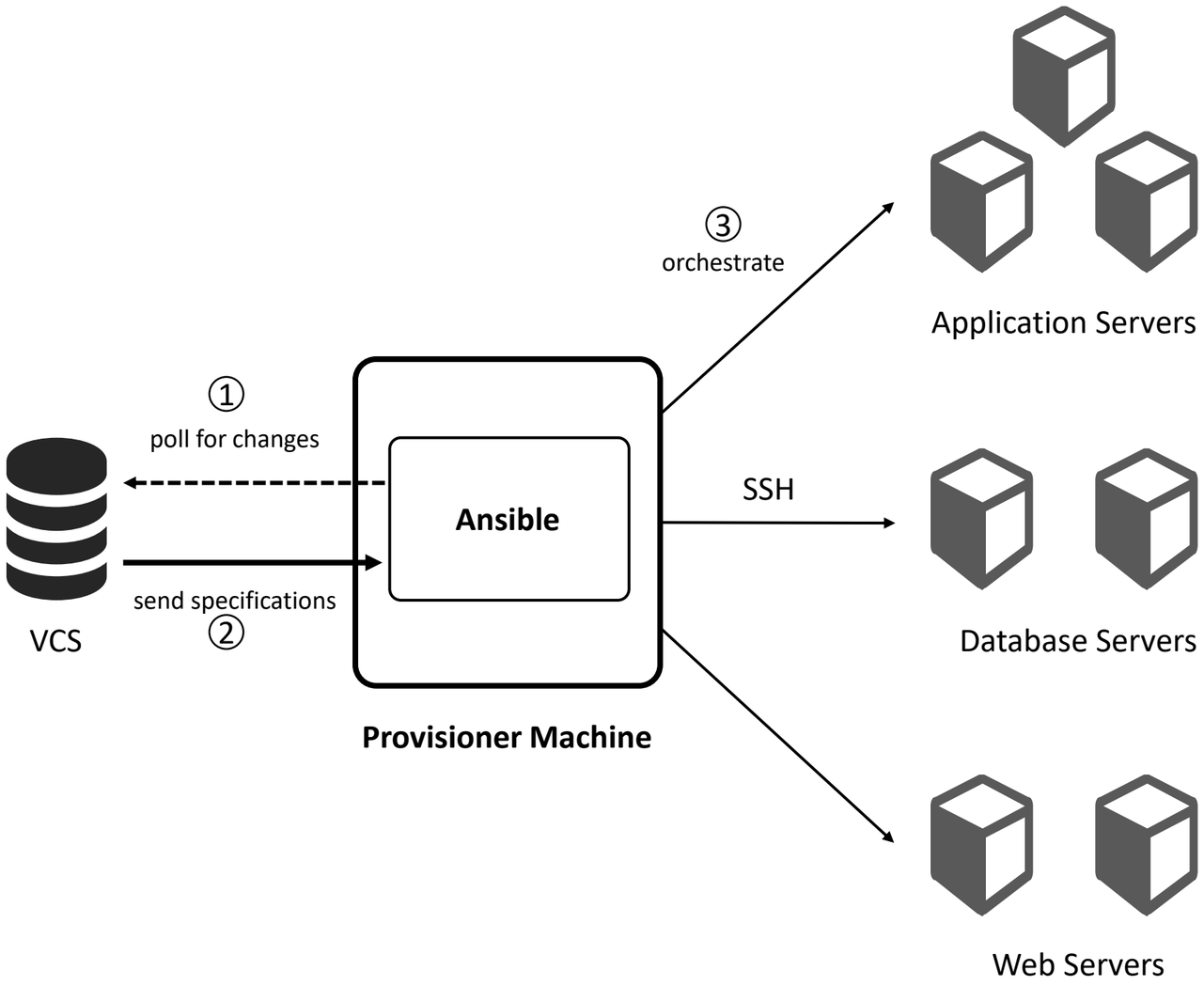
©Simplilearn. All rights reserved.

Ansible



©Simplilearn. All rights reserved.

Ansible Architecture



©Simplilearn. All rights reserved.

Ansible Installation Steps

```
root@ansible:~# apt-get install software-properties-common
Reading package lists... Done
Building dependency tree
Reading state information... Done
software-properties-common is already the newest version (0.96.24.32.5).
The following package was automatically installed and is no longer required:
  grub-pc-bin
Use 'apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 9 not upgraded.
root@ansible:~# apt-add-repository ppa:ansible/ansible
 Ansible is a radically simple IT automation platform that makes your applications and sy
om code to deploy and update your applications— automate in a language that approaches pl
remote systems.

http://ansible.com/
 More info: https://launchpad.net/~ansible/+archive/ubuntu/ansible
Press [ENTER] to continue or Ctrl-c to cancel adding it.

Hit:1 http://us-east1.gce.archive.ubuntu.com/ubuntu bionic InRelease
Hit:2 http://us-east1.gce.archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:3 http://us-east1.gce.archive.ubuntu.com/ubuntu bionic-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu bionic-security InRelease
Hit:5 http://archive.canonical.com/ubuntu bionic InRelease
Get:6 http://ppa.launchpad.net/ansible/ansible/ubuntu bionic InRelease [15.9 kB]
Get:7 http://ppa.launchpad.net/ansible/ansible/ubuntu bionic/main amd64 Packages [540 B]
Get:8 http://ppa.launchpad.net/ansible/ansible/ubuntu bionic/main Translation-en [344 B]
Fetched 16.8 kB in 1s (20.5 kB/s)
Reading package lists... Done
root@ansible:~# apt update
```

©Simplilearn. All rights reserved.

Ansible Installation Steps (Contd.)

```
root@ansible:~# apt install -y ansible
Reading package lists... Done
Building dependency tree
Reading state information... Done
ansible is already the newest version (2.7.1-1ppa~bionic).
The following package was automatically installed and is no longer required:
  grub-pc-bin
Use 'apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 9 not upgraded.
root@ansible:~# ansible --version
ansible 2.7.1
  config file = /etc/ansible/ansible.cfg
  configured module search path = [u'/root/.ansible/plugins/modules', u'/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python2.7/dist-packages/ansible
  executable location = /usr/bin/ansible
  python version = 2.7.15rc1 (default, Apr 15 2018, 21:51:34) [GCC 7.3.0]
root@ansible:~#
```

©Simplilearn. All rights reserved.

Ansible Installation Steps (Contd.)

```
root@ansible:~# apt install -y ansible
Reading package lists... Done
Building dependency tree
Reading state information... Done
ansible is already the newest version (2.7.1-1ppa~bionic).
The following package was automatically installed and is no longer required:
  grub-pc-bin
Use 'apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 9 not upgraded.
root@ansible:~# ansible --version
ansible 2.7.1
  config file = /etc/ansible/ansible.cfg
  configured module search path = [u'/root/.ansible/plugins/modules', u'/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python2.7/dist-packages/ansible
  executable location = /usr/bin/ansible
  python version = 2.7.15rc1 (default, Apr 15 2018, 21:51:34) [GCC 7.3.0]
root@ansible:~#
```

©Simplilearn. All rights reserved.

Ansible Installation Steps (Contd.)

SSH connectivity have to be set up between Ansible controller machine and node machines to manage configuration items.

- Generate SSH Key Pair on Ansible controller machine using the command “ssh-keygen –t rsa”.
- Now copy public key to “authorized_keys” file on other node machines.
- Once the key is copied, it can be verified using the SSH command. It should work without a password.

©Simplilearn. All rights reserved.

Ansible Installation Steps (Contd.)

- The same procedure needs to be repeated for all other nodes one by one.
- Once the setup is completed, the node's IP address should be mentioned in the Ansible inventory file.

```
root@ansible:~# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
```

```
root@ansible:~# ssh 10.142.0.3 "date"
Sat Nov 10 03:34:26 UTC 2018
```

©Simplilearn. All rights reserved.

Ansible Configuration and Example Commands

```
root@ansible:~# cat /etc/ansible/hosts
[servers]
10.142.0.3
10.142.0.2
root@ansible:~# ansible -m ping all
10.142.0.3 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
10.142.0.2 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
root@ansible:~# ansible -m shell -a "hostname" all
10.142.0.3 | CHANGED | rc=0 >>
node

10.142.0.2 | CHANGED | rc=0 >>
ansible

root@ansible:~# ansible -m shell -a "uname -a" servers
10.142.0.2 | CHANGED | rc=0 >>
Linux ansible 4.15.0-1023-gcp #24-Ubuntu SMP Wed Oct 10 13:28:59 UTC 2018 x86_64 x86_64 x86_64 GNU/Linux

10.142.0.3 | CHANGED | rc=0 >>
Linux node 4.15.0-1023-gcp #24-Ubuntu SMP Wed Oct 10 13:28:59 UTC 2018 x86_64 x86_64 x86_64 GNU/Linux

root@ansible:~#
```

©Simplilearn. All rights reserved.

Ansible Modules

Ansible has different types of module libraries. It can be used directly, or it can be included in playbooks. These modules are predefined source codes provided by Ansible to execute a set of steps. It’s easy to use Ansible modules as they do not need any coding experience.

- The modules use key-value arguments while managing resources.
- The modules can be used to manage services, packages, and files.

©Simplilearn. All rights reserved.

Ansible Modules (Contd.)

ansible-doc –l (Command to fetch all modules in Ansible)

ansible-doc <module_name> (Command to get documentation of a specific module in Ansible)

```
root@ansible:~# ansible-doc -l| head -n 4
a10_server                                      Manage A10 Networks AX/SoftAX/Thunder/vThunder devices' server object.
a10_server_axapi3                               Manage A10 Networks AX/SoftAX/Thunder/vThunder devices
a10_service_group                               Manage A10 Networks AX/SoftAX/Thunder/vThunder devices' service groups.
a10_virtual_server                              Manage A10 Networks AX/SoftAX/Thunder/vThunder devices' virtual servers.
ERROR! Unexpected Exception, this is probably a bug: [Errno 32] Broken pipe
root@ansible:~# ansible-doc -l| grep shell
shell                                           Execute commands in nodes.
vmware_vm_shell                                 Run commands in a VMware guest operating system
win_psmodule                                    Adds or removes a Powershell Module
win_shell                                       Execute shell commands on target hosts
root@ansible:~#
```

©Simplilearn. All rights reserved.

Ansible Modules (Contd.)

```
root@ansible:~# ansible-doc ping
> PING    (/usr/lib/python2.7/dist-packages/ansible/modules/system/ping.py)

        A trivial test module, this module always returns `pong' on successful contact. It does not make sense in playbooks, but it is useful from
        `/usr/bin/ansible' to verify the ability to login and that a usable Python is configured. This is NOT ICMP ping, this is just a trivial test module that
        requires Python on the remote-node. For Windows targets, use the [win_ping] module instead. For Network targets, use the [net_ping] module instead.

OPTIONS (= is mandatory):

- data
        Data to return for the `ping' return value.
        If this parameter is set to `crash', the module will cause an exception.
        [Default: pong]


NOTES:
      * For Windows targets, use the [win_ping] module instead.
      * For Network targets, use the [net_ping] module instead.

AUTHOR: Ansible Core Team, Michael DeHaan
        METADATA:
          status:
          - stableinterface
          supported_by: core


EXAMPLES:
# Test we can logon to 'webservers' and execute python with json lib.
# ansible webservers -m ping

# Example from an Ansible Playbook
- ping:

# Induce an exception to see what happens
- ping:
    data: crash

RETURN VALUES:


ping:
    description: value provided with the data parameter
    returned: success
    type: string
    sample: pong
```

©Simplilearn. All rights reserved.

Ansible Playbook

Ansible playbook is a YAML template used to manage system resources automatically. There is no need to login to each and every server. These templates can be scheduled for both, system configuration and application deployment.

Below is the example of one YAML template used to install and manage the NTP package on Ansible systems.

```
root@ansible:~# cat ntp.yaml
---
- hosts: all
  tasks:
    - name: Run the equivalent of "apt-get update" as a separate step
      apt: update_cache=yes
    - name: Install NTP package
      apt: name=ntp state=present
    - name: Restart NTP service
      service: name=ntp state=restarted
```

©Simplilearn. All rights reserved.

Ansible Playbook Execution

```
root@ansible:~# ansible-playbook ntp.yaml

PLAY [all] ***********************************************************************************************************************

TASK [Gathering Facts] ***********************************************************************************************************
ok: [10.142.0.2]
ok: [10.142.0.3]

TASK [Run the equivalent of "apt-get update" as a separate step] *****************************************************************
changed: [10.142.0.3]
changed: [10.142.0.2]

TASK [Install NTP package] *******************************************************************************************************
ok: [10.142.0.2]
ok: [10.142.0.3]

TASK [Restart NTP service] *******************************************************************************************************
changed: [10.142.0.2]
changed: [10.142.0.3]

PLAY RECAP ***********************************************************************************************************************
10.142.0.2                 : ok=4    changed=2    unreachable=0    failed=0
10.142.0.3                 : ok=4    changed=2    unreachable=0    failed=0

root@ansible:~# ansible -m shell -a "service ntp status | head -n 3" all
 [WARNING]: Consider using the service module rather than running service.  If you need to use command because service is insufficient you
can add warn=False to this command task or set command_warnings=False in ansible.cfg to get rid of this message.

10.142.0.2 | CHANGED | rc=0 >>
● ntp.service - Network Time Service
   Loaded: loaded (/lib/systemd/system/ntp.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2018-11-10 03:57:43 UTC; 12s ago

10.142.0.3 | CHANGED | rc=0 >>
● ntp.service - Network Time Service
   Loaded: loaded (/lib/systemd/system/ntp.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2018-11-10 03:57:43 UTC; 12s ago
```

©Simplilearn. All rights reserved.

Assisted Practice

Duration: 70 mins

Set Up Apache Web Server Using Ansible

Problem Statement: You are given a project to demonstrate the use of Ansible by setting up the Apache Web server.

Access: Click on the **Labs** tab on the left side panel of the LMS. Copy or note the username and password that is generated. Click on the **Launch Lab** button. On the page that appears, enter the username and password in the respective fields, and click **Login**.

©Simplilearn. All rights reserved.

Assisted Practice: Guidelines to Set Up Apache Web Server Using Ansible

1. Login to your Ubuntu Lab and open the terminal.
2. Install Ansible in Ubuntu.
3. Establish connection between Ansible controller and the node machine.
4. Write Ansible YAML script to install Ansible software.
5. Run Ansible YAML script.

©Simplilearn. All rights reserved.

Configuration Management Tools

Overview of SaltStack

©Simplilearn. All rights reserved.

SaltStack Configuration Management Tool

- SaltStack, like Ansible, is also a Python-based system provisioning tool.
- Tom Hatch, an IT Architect, built SaltStack in 2011. He was involved in both, Puppet and Chef configuration management tools.
- SaltStack is an open-source CM tool that comes under Apache 2.0 license.
- Like Ansible, YAML or Python-based scripts are used to manage system resources.

SALTSTACK

©Simplilearn. All rights reserved.

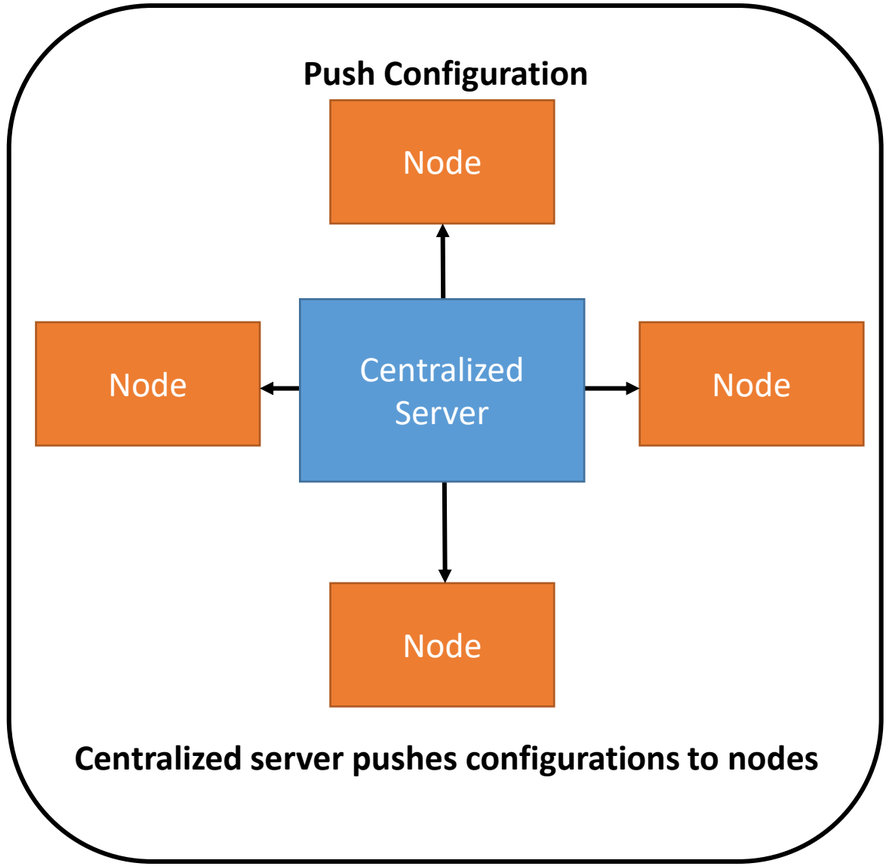
SaltStack Configuration Management Tool (Contd.)

- SaltStack is one of the CM tools competing with Chef, Puppet, and Ansible.
- Like Ansible, learning curve is less in case of SaltStack, as the same modules are here to manage system resources.
- SaltStack modules can be used to manage services, packages, and file resources.

©Simplilearn. All rights reserved.

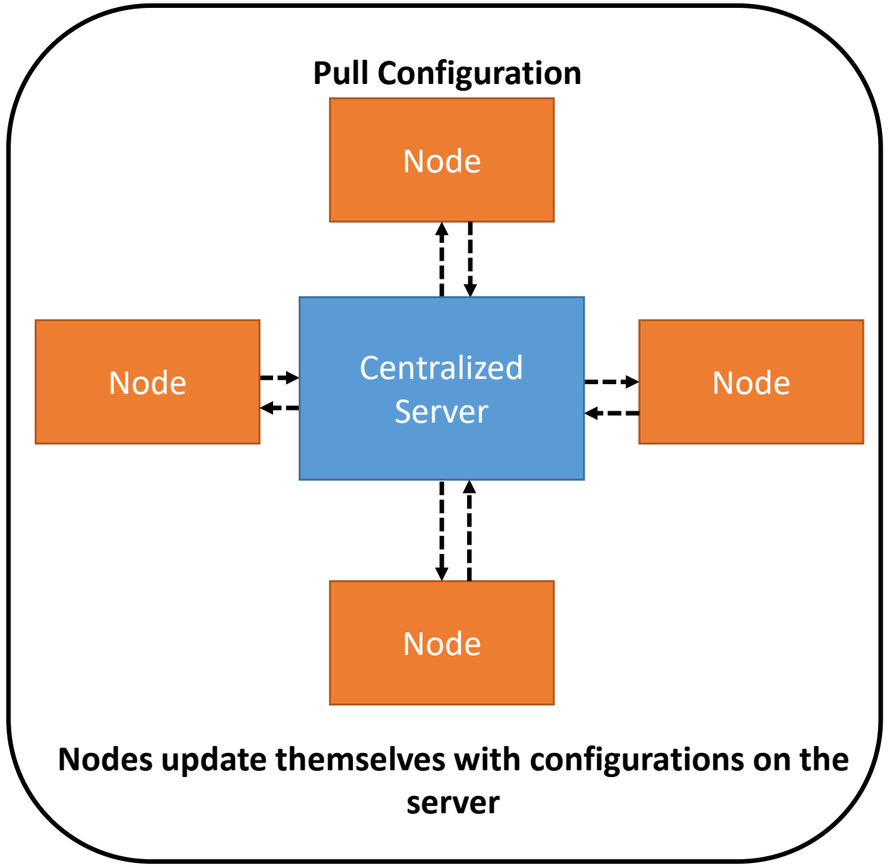
Architecture Similarity

Ansible, SaltStack



Centralized server pushes configurations to nodes

Puppet, Chef



Nodes update themselves with configurations on the server

©Simplilearn. All rights reserved.

Configuration Management Tools

Comparison of Ansible, Puppet, Chef, and SaltStack

©Simplilearn. All rights reserved.

simplilearn

Comparison of Configuration Management Tools

CM Tools	Pros	Cons	Enterprise annual cost for an average of 100 nodes
Ansible	• Simple architecture • Low learning curve	• No Windows support for controller machine • GUI is not good as compared to Chef and Puppet	Ansible open-source for 100 nodes → $0 Ansible Tower for 100 nodes → $3000 Ansible Tower Enterprise for 100 nodes → $10,000
Salt	• Scalable and fast • Easy to manage	• GUI is not good	Salt Enterprise for 100 nodes → $15,000
Chef	• More features • Perfect GUI with lot of features	• Needs knowledge of Ruby • Takes more time to understand • Learning curve is more in Chef • Only 10 nodes allowed in open-source license	Chef Enterprise for 100 nodes → $7200
Puppet	• Oldest tool in market • GUI with better features	• Difficult to understand and configure • Has less number of integrations with other tools	Puppet Enterprise for 100 nodes → $10,500

©Simplilearn. All rights reserved.

Key Takeaways

You are now able to:

- Explain the concepts of configuration management tools
- Demonstrate Puppet installation
- Demonstrate Chef installation
- Demonstrate Ansible installation and its usage
- Describe SaltStack
- Select the suitable CM tool for your organization

©Simplilearn. All rights reserved.

Knowledge Check

©Simplilearn. All rights reserved.

Knowledge Check 1	**Which configuration item cannot be managed using configuration management tool?**

a. Git Package

b. Server Bounce

c. Changing RAM hardware for server

d. Creating file on file system

©Simplilearn. All rights reserved.

Knowledge Check 1	**Which configuration item cannot be managed using configuration management tool?**

a. Git Package

b. Server Bounce

c. Changing RAM hardware for server

d. Creating file on file system

The correct answer is **c. Changing RAM hardware for server**

Only system-related configuration items can be managed using a CM tool. Hardware-related items can't be managed using these tools.

©Simplilearn. All rights reserved.

Knowledge Check 2	**What is the main objective of configuration management tools?**

a. Managing configuration items

b. Controlling modifications on server

c. Auditing configurations done to the system

d. All of the above

©Simplilearn. All rights reserved.

Knowledge Check 2	**What is the main objective of configuration management tools?**

a. Managing configuration items

b. Controlling modifications on server

c. Auditing configurations done to the system

d. All of the above

The correct answer is **d. All of the above**

All of the above options can be used to manage various configurations on the system.

©Simplilearn. All rights reserved.

Knowledge Check 3	**Which one of the following is a pull-based configuration management tool ?**

a. Chef

b. Ansible

c. Salt

d. Jaro

©Simplilearn. All rights reserved.

Knowledge Check 3	**Which one of the following is a pull-based configuration management tool ?**

a. Chef

b. Ansible

c. Salt

d. Jaro

The correct answer is **a. Chef**

Chef-client continuously pings the master node for configuration changes and pulls the configuration scripts.

©Simplilearn. All rights reserved.

Knowledge Check 4	**Which one of the following is an open-source configuration management tool ?**

a. Chef

b. Ansible

c. SourceForge

d. Puppet

©Simplilearn. All rights reserved.

Knowledge Check 4	**Which one of the following is an open-source configuration management tool ?**

a. Chef

b. Ansible

c. SourceForge

d. Puppet

The correct answer is **b. Ansible**

Ansible is an open-source software used to manage production environment.

©Simplilearn. All rights reserved.

Lesson-End Project

Set Up MySQL Database Using Ansible

Duration: 40 mins

Problem Statement:

Perform the following actions:

- Install Ansible in Ubuntu if it is not installed.
- Establish connection between Ansible controller and the node machine.
- Write Ansible YAML script to install Ansible software.
- Run Ansible YAML script.

Access: Click on the **Labs** tab on the left side panel of the LMS. Copy or note the username and password that is generated. Click on the **Launch Lab** button. On the page that appears, enter the username and password in the respective fields, and click **Login**.

©Simplilearn. All rights reserved.

Thank You

©Simplilearn. All rights reserved.

DevOps Certification Training

Lesson 06: Containerization with Docker

© Simplilearn. All rights reserved.

Learning Objectives

By the end of this lesson, you will be able to:

- Describe the importance of Docker over Virtual Machines
- Demonstrate on Docker images, Containers, and Docker Registry
- Demonstrate on Docker Compose and Docker Hub
- Describe the importance of Docker Networking

©Simplilearn. All rights reserved.

Containerization with Docker

Overview of Docker

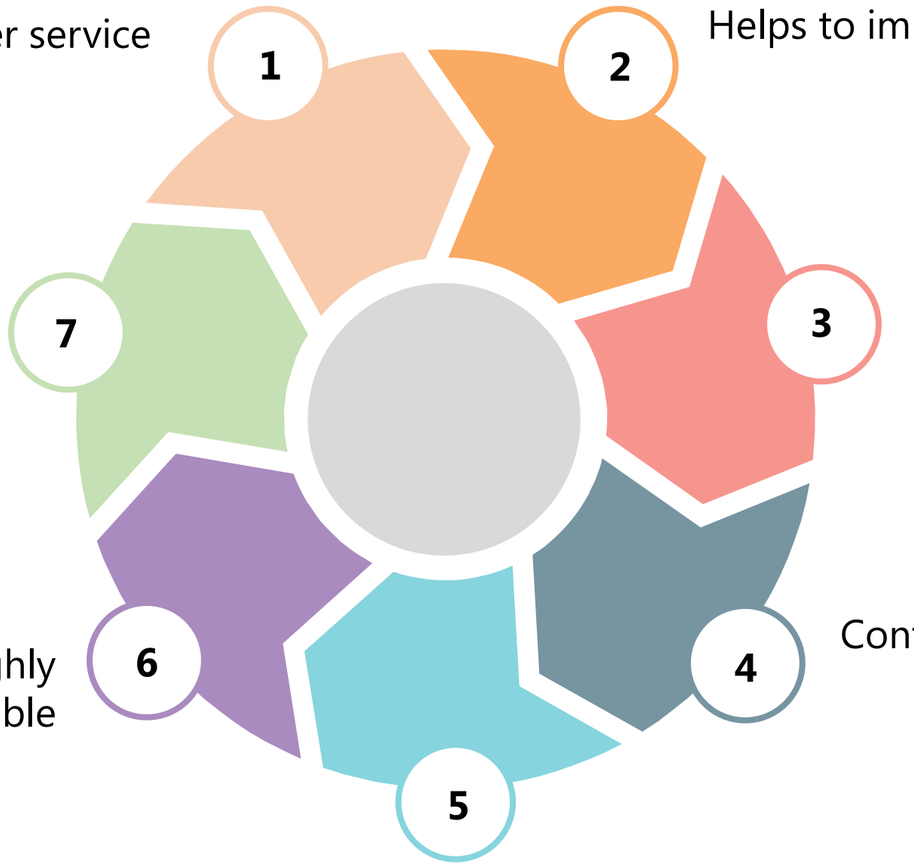
©Simplilearn. All rights reserved.

Docker Introduction

Docker is an open-source software that enables microservices and containerization of software applications. It's a light weight container service that consists of source code with dependencies bundled up together.

©Simplilearn. All rights reserved.

Docker Features



1. Light weight container service
2. Helps to implement micro services
3. Source codes with dependencies are bundled up together inside container
4. Containers runs in isolation
5. Helps to resolve the versioning problem with software
6. Fast, reliable and highly scalable
7. Is independent of the operating system and supports platforms based on both Windows and Unix

©Simplilearn. All rights reserved.

Docker Workflow

Pull or Build Docker image from Docker hub, bundle source with dependencies inside this Docker image.

Runs pulled Docker image on Docker host and start container on the host.

Container will be running in background and we can kill or stop container anytime we want to do so.

We can access applications hosted in Docker container using host machine IP and port details.

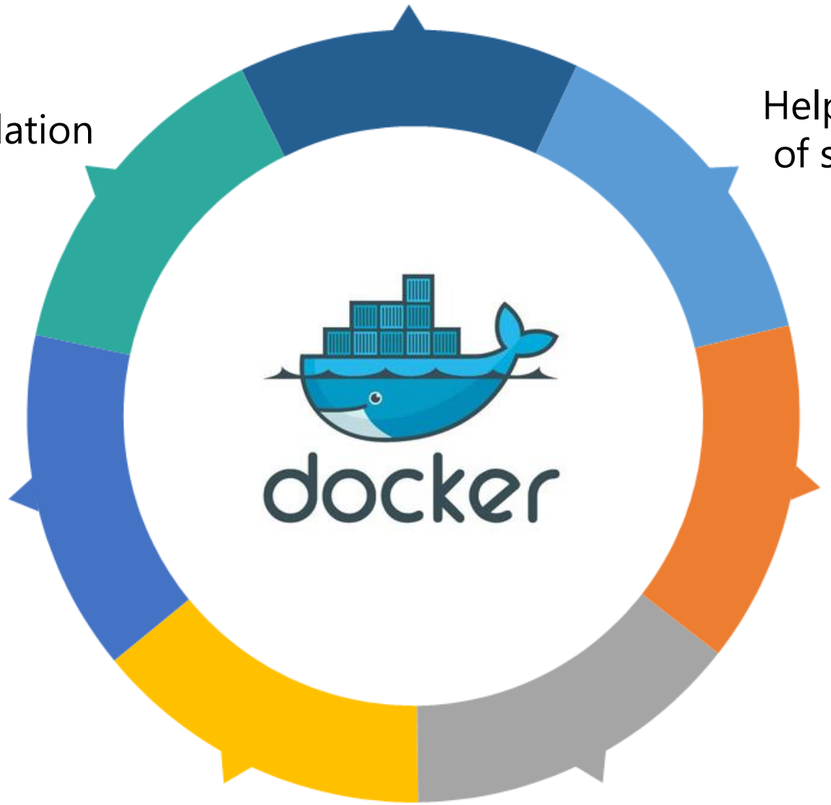
©Simplilearn. All rights reserved.

Docker Benefits

Infrastructure cost can be saved

Applications run in total isolation

Helps to establish standardization of software applications



Can be built once and deploy to multiple environments and operating systems

Eliminates versioning and dependency issues related to hosting source code on server

Is a better solution for CI/CD implementation of software applications

Helps in fast configurations and building applications

©Simplilearn. All rights reserved.

Containerization with Docker

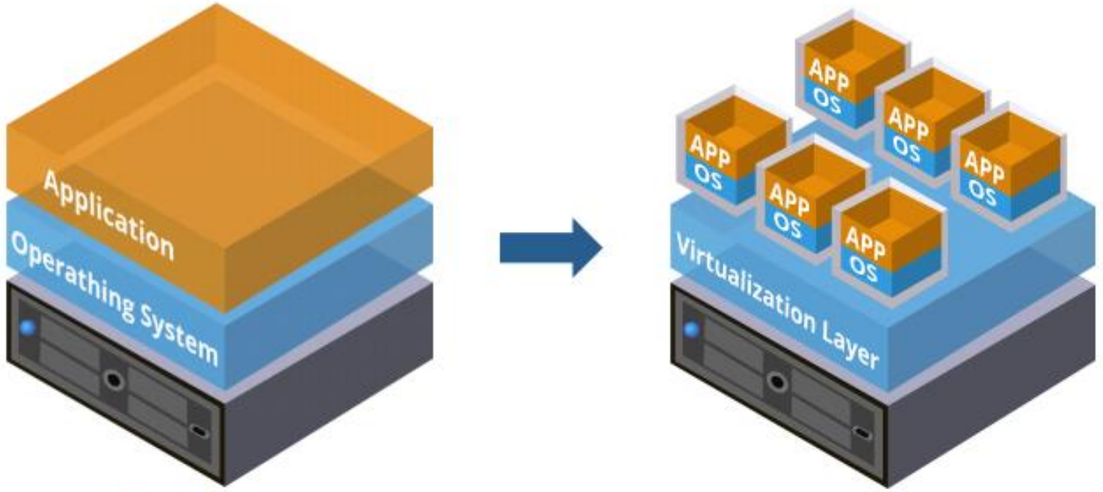
Overview of Virtualization

©Simplilearn. All rights reserved.

Virtualization

Virtualization refers to creating virtual resources like server, CPU, storage or network. You can create multiple sub-machines that can be used to host software applications.

- It helps to create multiple environment out of the same physical system
- It helps in optimal utilization of physical resources and helps in reducing expenses
- It helps to host multiple operating systems isolated from each other



Traditional Architecture

Virtual Architecture

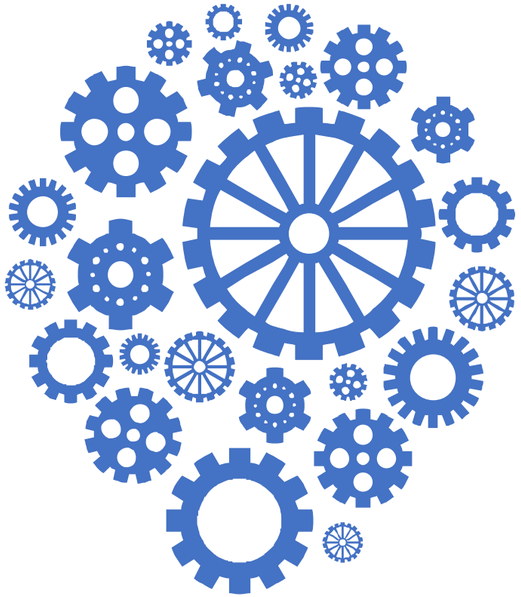
©Simplilearn. All rights reserved.

Virtualization Advantages

Creates multiple environments on the same physical system

Helps to reduce the infrastructure cost margin

Easy to clone virtual machines with software



Allow multiple machines run at same time in total isolation

Able to host different operating systems on the same physical machine

©Simplilearn. All rights reserved.

Comparison between Docker and Virtual Machines

	Docker	Virtual Machine
Operating System	Basic Operating system based images	Any
Startup Time	1-2 s	1-2 min
Disk Usage	Small	Large
Memory Usage	Small	Large
Configuration setup	Easy and simple	Complex
Hardware Requirement	Less	More
Security	Not Matured	Mature and Better

©Simplilearn. All rights reserved.

Containerization with Docker

Docker installation on Multiple OS

©Simplilearn. All rights reserved.

simplilearn

Docker on Linux

Docker was initially launched for Unix-based environment. Later, it was released for other platforms. Docker installation is available for Ubuntu, centos, Red hat and other Unix based platforms.

```
root@docker:~# apt update
Hit:1 http://us-east1.gce.archive.ubuntu.com/ubuntu bionic InRelease
Get:2 http://us-east1.gce.archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:3 http://us-east1.gce.archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Hit:4 http://archive.canonical.com/ubuntu bionic InRelease
Get:5 http://us-east1.gce.archive.ubuntu.com/ubuntu bionic-updates/main Sources [211 kB]
Get:6 http://us-east1.gce.archive.ubuntu.com/ubuntu bionic-updates/universe Sources [97.5 kB]
Get:7 http://us-east1.gce.archive.ubuntu.com/ubuntu bionic-updates/multiverse Sources [3212 B]
Get:8 http://us-east1.gce.archive.ubuntu.com/ubuntu bionic-updates/main amd64 Packages [437 kB]
Get:9 http://us-east1.gce.archive.ubuntu.com/ubuntu bionic-updates/main Translation-en [164 kB]
Get:10 http://us-east1.gce.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 Packages [575 kB]
Get:11 http://us-east1.gce.archive.ubuntu.com/ubuntu bionic-updates/universe Translation-en [154 kB]
Get:12 http://us-east1.gce.archive.ubuntu.com/ubuntu bionic-updates/multiverse amd64 Packages [5696 B]
Get:13 http://security.ubuntu.com/ubuntu bionic-security InRelease [83.2 kB]
Get:14 http://security.ubuntu.com/ubuntu bionic-security/main Sources [61.0 kB]
Get:15 http://security.ubuntu.com/ubuntu bionic-security/universe Sources [23.0 kB]
Get:16 http://security.ubuntu.com/ubuntu bionic-security/multiverse Sources [1332 B]
Get:17 http://security.ubuntu.com/ubuntu bionic-security/main amd64 Packages [205 kB]
Get:18 http://security.ubuntu.com/ubuntu bionic-security/main Translation-en [80.9 kB]
Get:19 http://security.ubuntu.com/ubuntu bionic-security/universe amd64 Packages [96.6 kB]
Get:20 http://security.ubuntu.com/ubuntu bionic-security/universe Translation-en [54.6 kB]
Get:21 http://security.ubuntu.com/ubuntu bionic-security/multiverse amd64 Packages [1448 B]
Fetched 2417 kB in 1s (2053 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
20 packages can be upgraded. Run 'apt list --upgradable' to see them.
root@docker:~# apt install docker.io
Reading package lists... Done
Building dependency tree
Reading state information... Done
docker.io is already the newest version (18.06.1-0ubuntu1~18.04.1).
The following package was automatically installed and is no longer required:
  grub-pc-bin
Use 'apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 20 not upgraded.
root@docker:~#
```

©Simplilearn. All rights reserved.

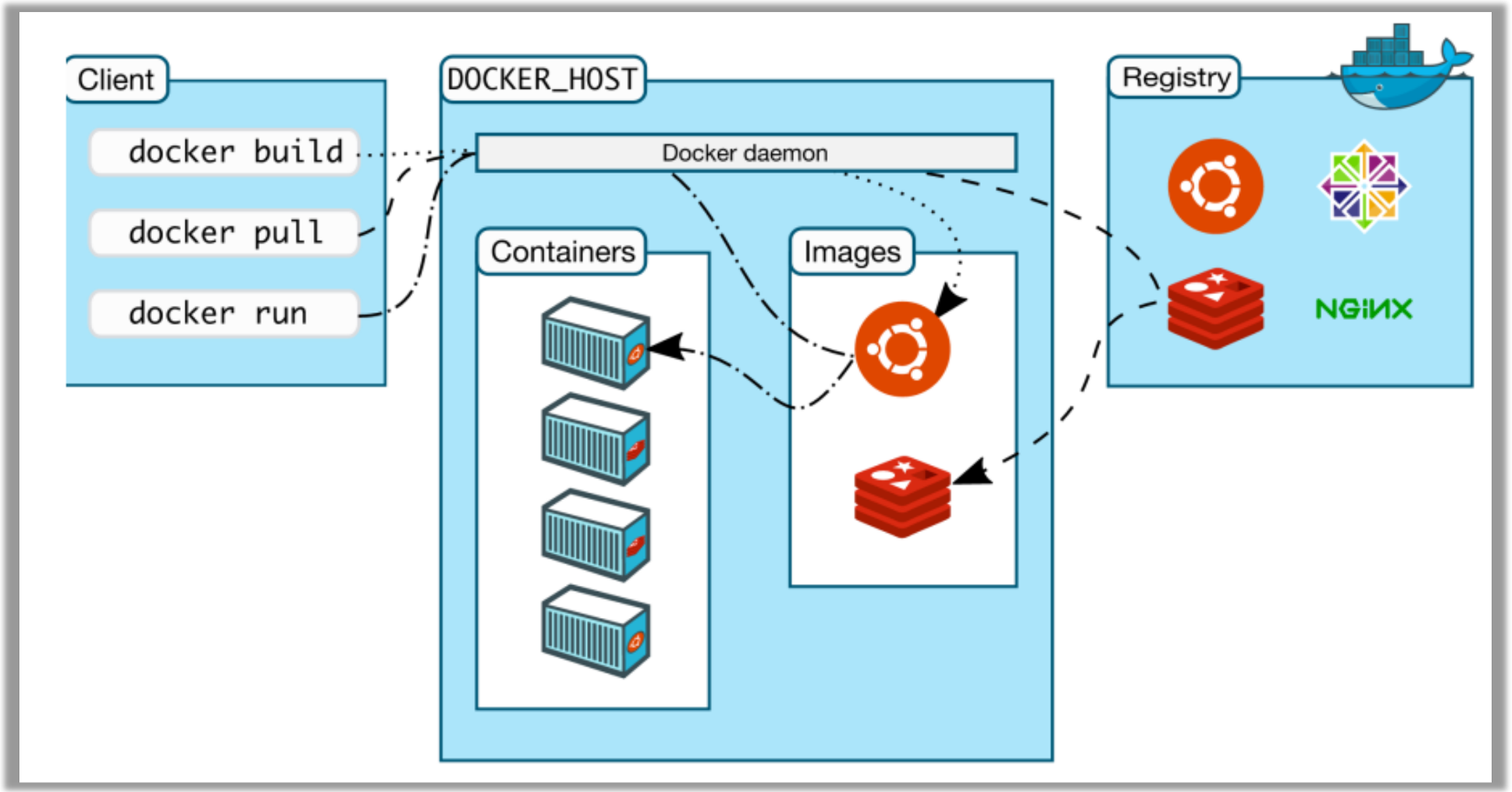
Docker on Desktop

- Supports desktop installations for Mac and Windows platforms
- Availability of .DMG and .EXE file support for Mac and Windows respectively
- Requires minimum 4 GB of RAM for both Mac and Windows
- Needs xhyve hypervisor in Mac and Hyper V in Windows to implement Docker
- Windows starter and Windows home versions are not supported by Docker

©Simplilearn. All rights reserved.

Docker Architecture

Docker is lightweight container service and the architecture of Docker is quiet simple. Docker follow client-server architecture.



©Simplilearn. All rights reserved.

Docker Architecture (Contd.)

- Docker Client
  - It is a command-line interface connected to Docker daemon using REST API.
  - It is used to manage images and containers on Docker host.
- Docker Daemon
  - Docker Daemon is kind of background process running on server and responsible for keeping applications and Docker images up and running.
  - In case, it is down, all Docker-hosted applications will also be down.

©Simplilearn. All rights reserved.

Docker Architecture (Contd.)

- Docker Registry
  - Registry is used to store public and private Docker images.
  - You can also download already published Docker images from Docker Hub.
  - You can pull enterprise Docker images from Docker Store such as Oracle, IBM Web Sphere or other proprietary software.

©Simplilearn. All rights reserved.

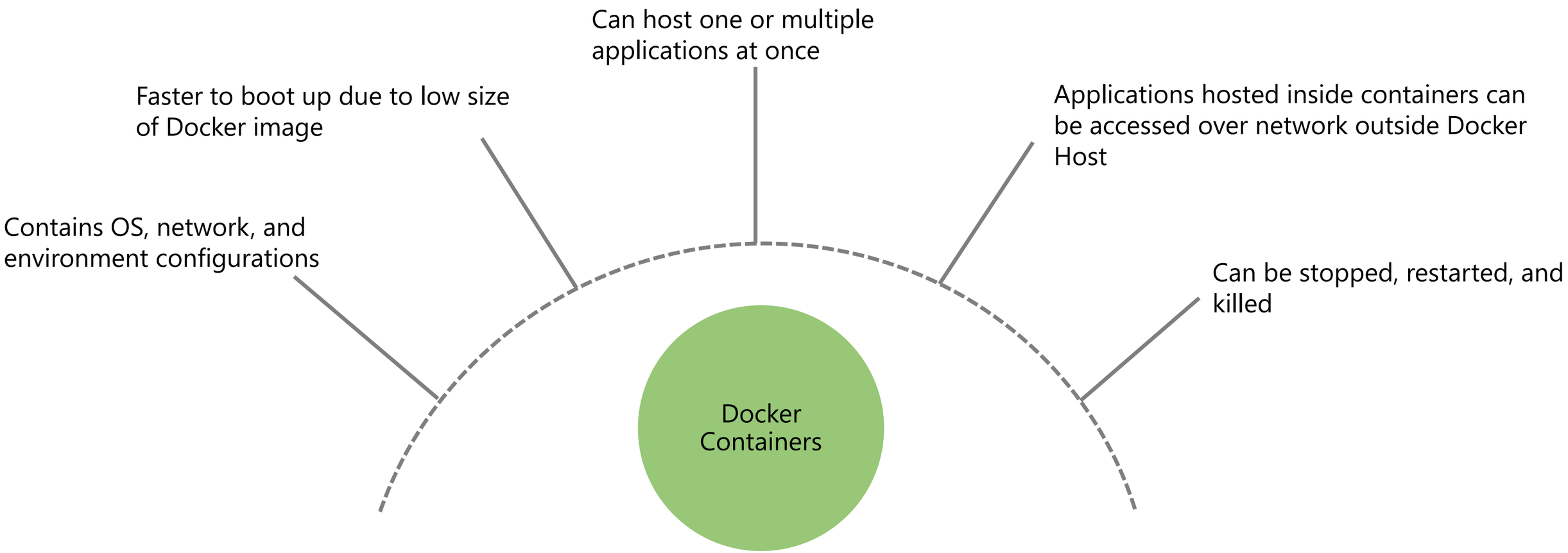
Docker Images

1. Read only template combination of multiple layers
2. Comprises of basic Operating System binaries
3. We can store Docker images on Docker hub or private repositories
4. Hosts software applications, web containers, application servers, and even databases
5. Exists for various open source tools in Docker hub and its free to pull those images
6. Images can be Built using custom Dockerfile
7. Smaller in size which helps to store and run Docker images easily

©Simplilearn. All rights reserved.

Docker Containers

An instance of Docker image is called Docker Container. Docker containers host different types of applications inside them.



©Simplilearn. All rights reserved.

Execution on Docker Images

- The command to check all existing Docker images on Docker hosts is “docker images”.
- The command to pull a Docker image from either Docker hub or private repository is docker pull “docker pull <image_name>:<tag>”.
- You can get the image name from Docker hub or you can also search it using the command “docker search <image_name>”.
- Docker pull command will first check the image on local system. In case it is not found there, Docker image will be pulled from Docker hub.

©Simplilearn. All rights reserved.

Execution on Docker Images (Contd.)

```
root@docker:~# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
root@docker:~# docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
473ede7ed136: Pull complete
c46b5fa4d940: Pull complete
93ae3df89c92: Pull complete
6b1eed27cade: Pull complete
Digest: sha256:29934af957c53004d7fb6340139880d23fb1952505a15d69a03af0d1418878cb
Status: Downloaded newer image for ubuntu:latest
root@docker:~# docker pull ubuntu:18.04
18.04: Pulling from library/ubuntu
Digest: sha256:29934af957c53004d7fb6340139880d23fb1952505a15d69a03af0d1418878cb
Status: Downloaded newer image for ubuntu:18.04
root@docker:~# docker pull ubuntu:18.10
18.10: Pulling from library/ubuntu
72f45ff89b78: Pull complete
9f034a33b165: Pull complete
386fee7ab4d3: Pull complete
f941b4ac6aa8: Pull complete
Digest: sha256:ef5d78b10fcada65484b2dd9cd6e0704fba36021a101b2b52023a1566c4ebad4
Status: Downloaded newer image for ubuntu:18.10
root@docker:~# docker pull centos
Using default tag: latest
latest: Pulling from library/centos
aeb7866da422: Pull complete
Digest: sha256:67dad89757a55bfdfabec8abd0e22f8c7c12a185651472647022806 3ed86593b
Status: Downloaded newer image for centos:latest
root@docker:~# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
ubuntu              18.10               7e10e8cb09ba        3 weeks ago         73.7MB
ubuntu              18.04               ea4c82dcd15a        3 weeks ago         85.8MB
ubuntu              latest              ea4c82dcd15a        3 weeks ago         85.8MB
centos              latest              75835a67d134        5 weeks ago         200MB
root@docker:~#
```

©Simplilearn. All rights reserved.

Execution on Docker Images (Contd.)

- You can remove Docker image using command below:

  docker rmi <image_name>:<tag>

- Docker images consumes local disk space, you can have to continuously work on maintenance.

- You can also export Docker images to tar file so that they can be transferred from one server to another.

  docker save <image_name>:<tag> -o <tar_filename>

- Docker image can be imported with all layers from tar file using command below. All previous details regarding image will be loaded automatically.

  docker load –I <tar_filename>

©Simplilearn. All rights reserved.

Execution on Docker Images (Contd.)

```
root@docker:~# docker images
oREPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
ubuntu              18.10               7e10e8cb09ba        3 weeks ago         73.7MB
ubuntu              18.04               ea4c82dcd15a        3 weeks ago         85.8MB
ubuntu              latest              ea4c82dcd15a        3 weeks ago         85.8MB
centos              latest              75835a67d134        5 weeks ago         200MB
root@docker:~# docker rmi ubuntu
Untagged: ubuntu:latest
root@docker:~# docker rmi ubuntu:18.04
Untagged: ubuntu:18.04
Untagged: ubuntu@sha256:29934af957c53004d7fb6340139880d23fb1952505a15d69a03af0d1418878cb
Deleted: sha256:ea4c82dcd15a33e3e9c4c37050def20476856a08e59526fbe533cc4e98387e39
Deleted: sha256:2ac9356b41d2d032dc980b6ee2b2a911790a47b59b4fcfd92e52a0729a389403
Deleted: sha256:f19c7e29a7e3fbaf44997bab14791e3d3d26d689bc3e2e720a9d5d8a77f68d6c
Deleted: sha256:b951bb1959dc1a5dfeda46229b36b33c4cc01e3f682ee1b77af8dab1cc7cf8a3
Deleted: sha256:102645f1cf722254bbfb7135b524db45fbbac400e79e4d54266c000a5f5bc400
root@docker:~# docker save centos -o centos.tar
root@docker:~# docker rmi centos
Untagged: centos:latest
Untagged: centos@sha256:67dad89757a55bfdfabec8abd0e22f8c7c12a1856514726470228063ed86593b
Deleted: sha256:75835a67d1341bdc7f4cc4ed9fa1631a7d7b6998e9327272afea342d90c4ab6d
Deleted: sha256:f972d139738dfcd1519fd2461815651336ee25a8b54c358834c50af094bb262f
root@docker:~# docker load -i centos.tar
f972d139738d: Loading layer [==================================================>]  208.8MB/208.8MB
Loaded image: centos:latest
root@docker:~# docker images | grep centos
centos              latest              75835a67d134        5 weeks ago         200MB
root@docker:~#
```

©Simplilearn. All rights reserved.

Execution on Docker Containers

- The ‘run’ command is used to create an instance of Docker image.

  docker run -it <image_name> bash

  - -i or --interactive keeps STDIN open
  - -t or --tty allocates a pseudo TTY

- You can also run Docker image in detached mode using flag ‘d’.

  docker run -d <image_name>

- Docker container can be easily removed from Docker host using command below:

  docker rm <container_id> -f

- Docker container can be stopped, killed, and restarted using commands below:

  docker stop <container_id>
  docker restart <container_id>
  docker kill <container_id>

©Simplilearn. All rights reserved.

Execution on Docker Containers (Contd.)

```
root@docker:~# docker run -it centos bash
[root@4a5119324fbd /]# uname -a
Linux 4a5119324fbd 4.15.0-1023-gcp #24-Ubuntu SMP Wed Oct 10 13:28:59 UTC 2018 x86_64 x86_64 x86_64 GNU/Linux
[root@4a5119324fbd /]# exit
exit
root@docker:~# docker run -d jenkins
47b089a3c311f34d72e3f65b2d18b7e2f2b8663aedbb8779c80b78bd48f3f32e
root@docker:~# docker ps -a
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS                      PORTS
47b089a3c311        jenkins             "/bin/tini -- /usr/l…"   10 seconds ago      Up 10 seconds               8080/tcp, 5
4a5119324fbd        centos              "bash"                   36 seconds ago      Exited (0) 29 seconds ago
ky
0f896a8a9c81        centos              "bash"                   5 minutes ago       Exited (0) 5 minutes ago
root@docker:~# docker rm 47b089a3c311 4a5119324fbd 0f896a8a9c81 -f
47b089a3c311
4a5119324fbd
0f896a8a9c81
root@docker:~# docker run -d jenkins
5a149d38a29d7673b75624a147acec3deb3371d98cb55369e34181cf614a65fd
root@docker:~# docker stop 5a149d38a29d7673b75624a147acec3deb3371d98cb55369e34181cf614a65fd
5a149d38a29d7673b75624a147acec3deb3371d98cb55369e34181cf614a65fd
root@docker:~# docker ps -a
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS                        PORTS
5a149d38a29d        jenkins             "/bin/tini -- /usr/l…"   26 seconds ago      Exited (143) 10 seconds ago
root@docker:~# docker start 5a149d38a29d
5a149d38a29d
root@docker:~# docker ps -a
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS
5a149d38a29d        jenkins             "/bin/tini -- /usr/l…"   48 seconds ago      Up 4 seconds        8080/tcp, 50000/tcp
root@docker:~#
```

©Simplilearn. All rights reserved.

Execution on Docker Containers (Contd.)

- You can connect back to Docker container running in detached mode using command below:

  docker exec -it <container_id> bash

  - -i or --interactive keeps STDIN open
  - -t or --tty allocates a pseudo TTY

- Docker container can be paused and unpaused using the commands below:

  docker pause <container_id>

  docker unpause <container_id>

©Simplilearn. All rights reserved.

Execution on Docker Containers (Contd.)

```
root@docker:~# docker ps -a
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS
5a149d38a29d        jenkins             "/bin/tini -- /usr/l…"   8 minutes ago       Up 7 minutes        8080/tcp, 50000/tcp
root@docker:~# docker exec -it 5a149d38a29d bash
jenkins@5a149d38a29d:/$ ps -ef
UID        PID  PPID  C STIME TTY          TIME CMD
jenkins      1     0  0 15:36 ?        00:00:00 /bin/tini -- /usr/local/bin/jenkins.sh
jenkins      6     1  5 15:36 ?        00:00:28 java -jar /usr/share/jenkins/jenkins.war
jenkins     14     6  0 15:36 ?        00:00:00 [jenkins.sh] <defunct>
jenkins     72     0  0 15:44 pts/0    00:00:00 bash
jenkins     79    72  0 15:44 pts/0    00:00:00 ps -ef
jenkins@5a149d38a29d:/$ exit
exit
root@docker:~# docker pause 5a149d38a29d
5a149d38a29d
root@docker:~# docker ps -a
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS                  PORTS
5a149d38a29d        jenkins             "/bin/tini -- /usr/l…"   9 minutes ago       Up 8 minutes (Paused)   8080/tcp, 50000/t
root@docker:~# docker unpause 5a149d38a29d
5a149d38a29d
root@docker:~# docker ps -a
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS
5a149d38a29d        jenkins             "/bin/tini -- /usr/l…"   9 minutes ago       Up 8 minutes        8080/tcp, 50000/tcp
root@docker:~#
```

©Simplilearn. All rights reserved.

Assisted Practice

Duration: 30 mins

MySQL Database in Docker Container

Problem Statement: You are given a project to demonstrate the execution of MySQL database in Docker container.

Access: Click on the **Labs** tab on the left side panel of the LMS. Copy or note the username and password that is generated. Click on the **Launch Lab** button. On the page that appears, enter the username and password in the respective fields, and click **Login**.

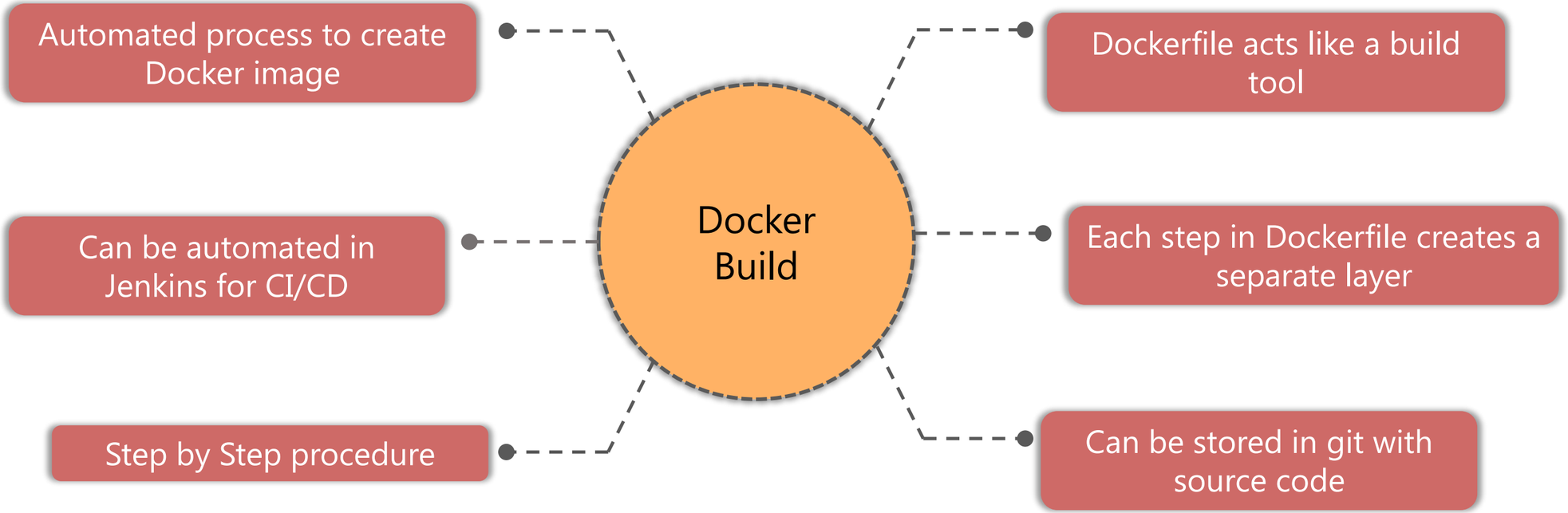
©Simplilearn. All rights reserved.

Assisted Practice: Guidelines to Demonstrate MySQL in Docker

1. Login to your Ubuntu Lab and open the terminal.
2. Create a folder to store all the files generated during the demonstration.
3. Clone the repository from Git and navigate inside the directory.
4. Grant the read, write, and execute permissions to the “runserver_first” file.
5. Execute the client and log the data in the database.
6. Execute the MySQL database command and enter the login password to use it.
7. Explore the MySQL database and confirm that it works efficiently.
8. Exit the MySQL Client.

©Simplilearn. All rights reserved.

Docker Automated Build



©Simplilearn. All rights reserved.

Dockerfile

- Dockerfile is used to build custom Docker images. You can add the application source or configure and install any application software in Docker image.
- There are a few attributes that can be used in Dockerfile:
  - FROM → Used to define base image information
  - MAINTAINER → Used to display the name and e-mail details of build owner
  - SHELL → Used to set the default shell
  - COPY → Used to transfer the local file to Docker container

©Simplilearn. All rights reserved.

Dockerfile (Contd.)

- ADD → This is similar to COPY, but you can use URL instead of local file. Archived files get unachieved automatically once transferred inside container to a specific folder
- RUN → Run command used to execute command inside Docker image
- ENV → Used to create environment variables inside Docker image
- COMMAND → Command defines the executable to be execute while running Docker image
- WORKDIR → Used to configure working DIR during executing Docker build command
- EXPOSE → Exposes port on which application will be hosted

©Simplilearn. All rights reserved.

Dockerfile (Contd.)

```
root@docker:~# docker build -t docker_apache .
Sending build context to Docker daemon  212.2MB
Step 1/12 : FROM ubuntu:18.04
 ---> ea4c82dcd15a
Step 2/12 : MAINTAINER SimpliLearn
 ---> Using cache
 ---> 70ac7fca0837
Step 3/12 : RUN apt-get update && apt-get install -y apache2 && apt-get clean && rm -rf /var/lib/apt/lists/*
 ---> Using cache
 ---> 1fc155264ef1
Step 4/12 : ENV APACHE_RUN_USER www-data
 ---> Using cache
 ---> 91f16ae7f5a8
Step 5/12 : ENV APACHE_RUN_GROUP www-data
 ---> Using cache
 ---> bb40bb196433
Step 6/12 : ENV APACHE_LOG_DIR /var/log/apache2
 ---> Using cache
 ---> a461eef9c4e0
Step 7/12 : WORKDIR /var/www/html
 ---> Using cache
 ---> 9ae2833a4176
Step 8/12 : COPY index.html /var/www/html
 ---> Using cache
 ---> cf7fafb6b846
Step 9/12 : ADD *.html /var/www/html
 ---> Using cache
 ---> e613dd7ed4d4
Step 10/12 : EXPOSE 80
 ---> Using cache
 ---> b8dd45cb6cb5
Step 11/12 : SHELL ["/bin/sh", "-c"]
 ---> Using cache
 ---> 33bbc46aa9de
Step 12/12 : CMD ["/usr/sbin/apache2", "-D", "FOREGROUND"]
 ---> Using cache
 ---> eb2d5063ce91
Successfully built eb2d5063ce91
Successfully tagged docker_apache:latest
root@docker:~# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
docker_apache       latest              eb2d5063ce91        12 seconds ago      182MB
ubuntu              18.04               ea4c82dcd15a        3 weeks ago         85.8MB
jenkins             latest              cd14cecfdb3a        4 months ago        696MB
root@docker:~#
```

©Simplilearn. All rights reserved.

Dockerfile Build

- Dockerfile is used to create custom Docker image using Docker build command.
- Docker build command follows syntax below:

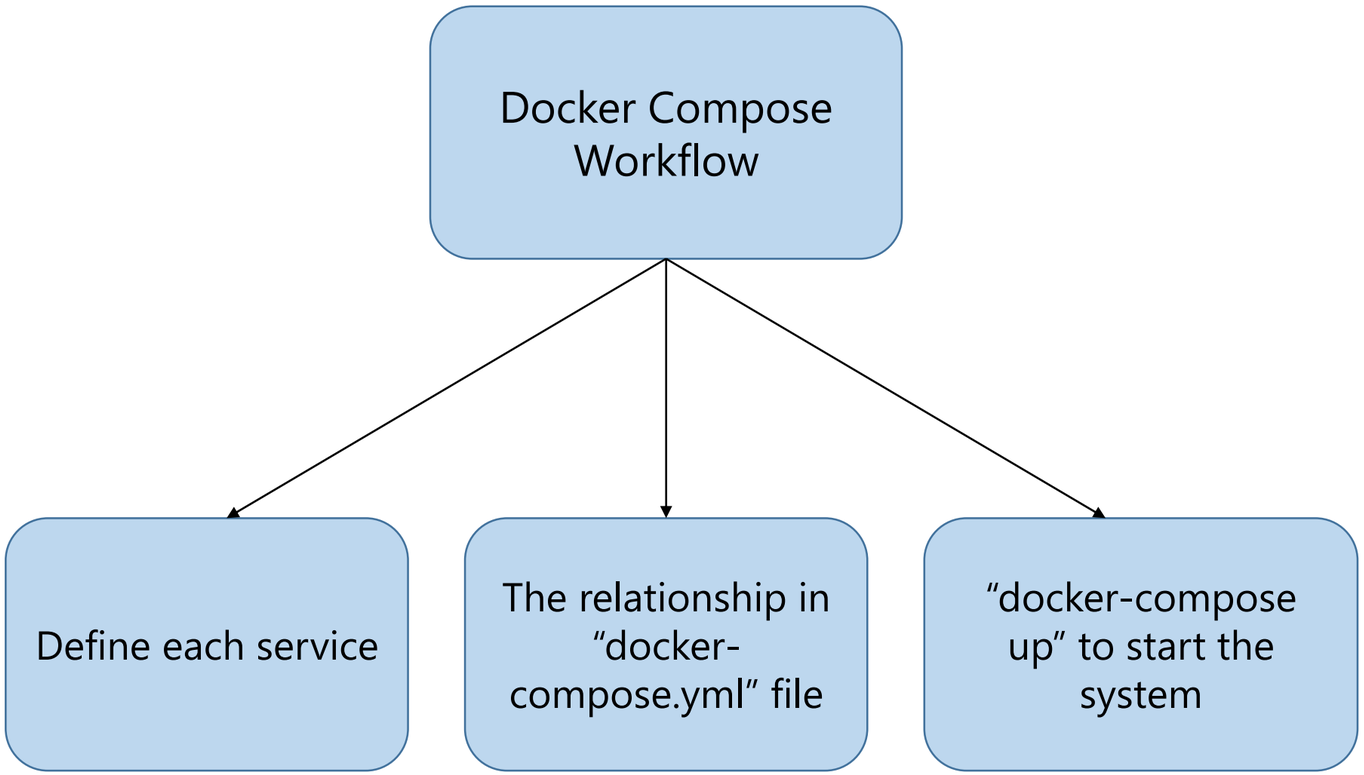
  docker build –t <image_name>:<tag> .
- This command creates an intermediate container, and that container will automatically be removed once image build is complete.

```
root@docker:~# cat Dockerfile
FROM ubuntu:18.04

MAINTAINER SimpliLearn

RUN apt-get update && apt-get install -y apache2 && apt-get clean && rm -rf /var/lib/apt/lists/*

ENV APACHE_RUN_USER www-data
ENV APACHE_RUN_GROUP www-data
ENV APACHE_LOG_DIR /var/log/apache2

WORKDIR /var/www/html

COPY index.html /var/www/html
ADD *.html /var/www/html

EXPOSE 80

SHELL ["/bin/sh", "-c"]
CMD ["/usr/sbin/apache2", "-D", "FOREGROUND"]
root@docker:~#
```

©Simplilearn. All rights reserved.

Docker Compose

- Docker Compose is used to define and run complex applications in Docker.
- The main functions of Docker Compose are building images, scaling containers, and rerunning containers that have stopped.
- It has a similar set of commands like Docker.
- Entire configuration can be stored and versioned along with the project.



©Simplilearn. All rights reserved.

Assisted Practice

Duration: 30 mins

Using Docker Compose to Manage a Container

Problem Statement: You are given a project to demonstrate the use of Docker Compose to manage a container.

Access: Click on the **Labs** tab on the left side panel of the LMS. Copy or note the username and password that is generated. Click on the **Launch Lab** button. On the page that appears, enter the username and password in the respective fields, and click **Login**.

©Simplilearn. All rights reserved.

Assisted Practice: Guidelines to Demonstrate the Use of Docker Compose

1. Login to your Ubuntu Lab and open the terminal.
2. Create a Dockerfile and build the Docker image with it.
3. Run the image and connect to the SSH server.
4. Exit the container once you confirm that you have logged in.
5. Create a “docker-compose.yml” file and connect to the server.
6. Stop the server and exit the container.

©Simplilearn. All rights reserved.

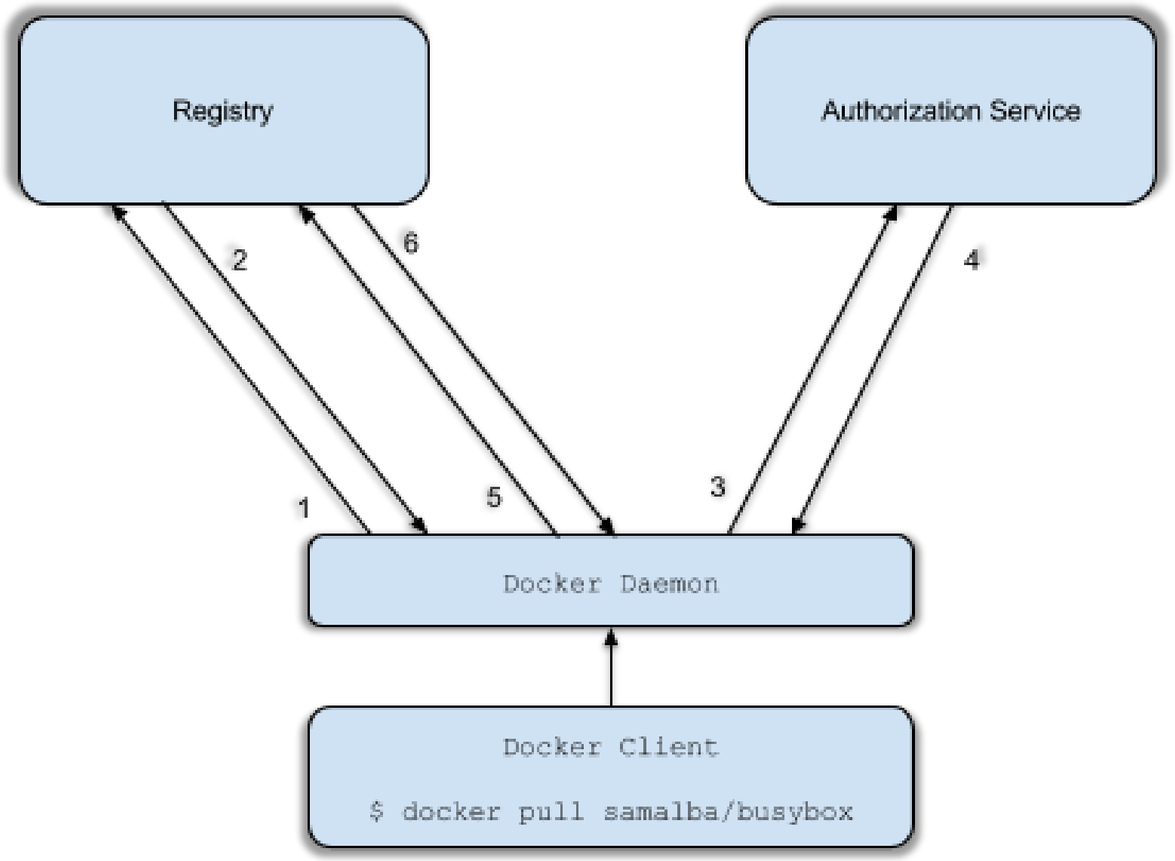
Containerization with Docker

Docker Registry

©Simplilearn. All rights reserved.

Docker Registry

- Docker Registry helps you share Docker images.
- You can also have different collaborators to a single Docker image.
- Both public and private both Docker images can be uploaded to Docker hub.
- Docker store consists of enterprise and trusted Docker images.



©Simplilearn. All rights reserved.

Docker Registry (Contd.)

- You can upload various versions of the same Docker image.
- The free account of Docker hub supports unlimited public Docker images and only one private Docker image.
- Some of the private Docker Registries, other than Docker hub and Docker store, are listed as below:
  - Google Container Registry
  - Amazon ECR
  - Jfrog Artifactory
  - Nexus Artifactory

©Simplilearn. All rights reserved.

Assisted Practice

Duration: 30 mins

Run Docker Registry with Centos

Problem Statement: You are given a project to demonstrate and run a Docker registry from an official registry image.

Access: Click on the **Labs** tab on the left side panel of the LMS. Copy or note the username and password that is generated. Click on the **Launch Lab** button. On the page that appears, enter the username and password in the respective fields, and click **Login**.

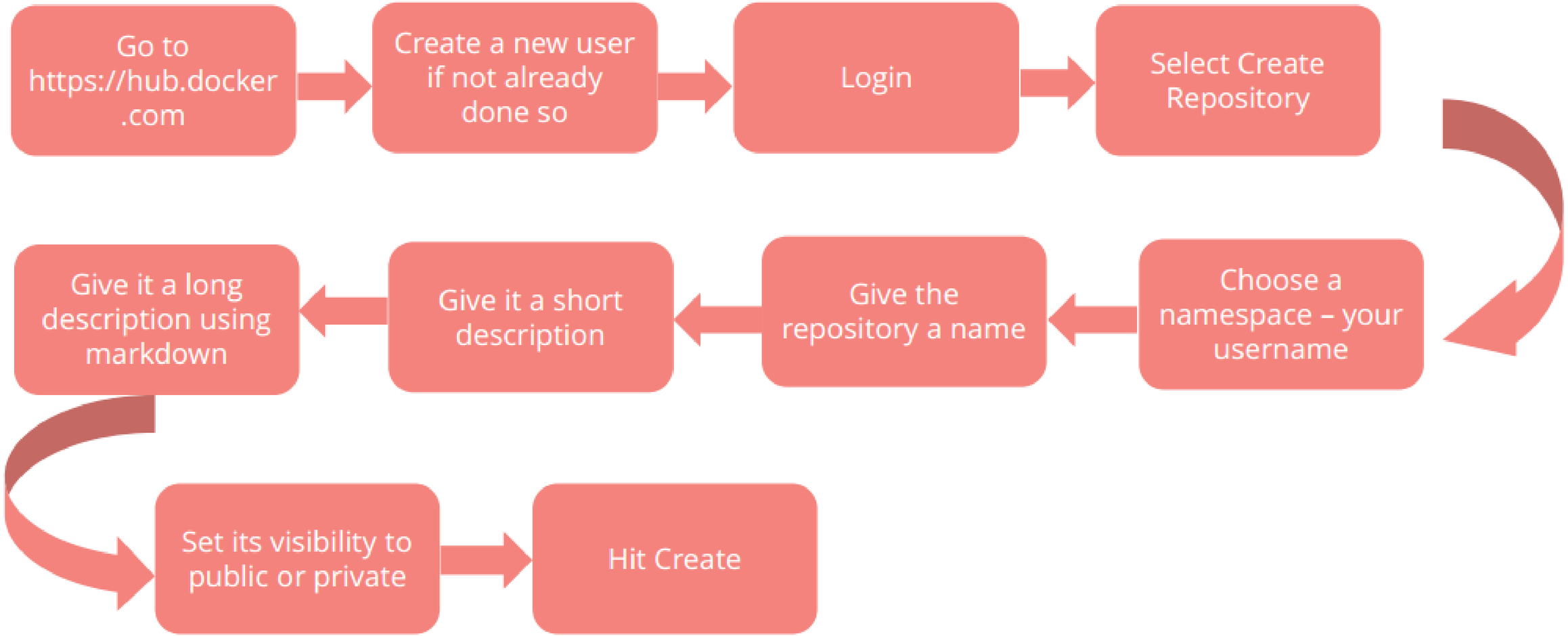
©Simplilearn. All rights reserved.

Assisted Practice: Guidelines to Run Docker Registry with Centos

1. Login to your Ubuntu Lab and open the terminal.
2. Pull a recent version of the Centos Linux container.
3. Run the registry in a new Docker container. Pull a new image from Docker Hub.
4. Push the image to the local registry.
5. Remove the image from the local cache.
6. Pull the image from the local repository.
7. Run the new Docker image.
8. Exit the container once the image is confirmed.

©Simplilearn. All rights reserved.

Docker Hub Configuration: Step by Step Procedure



©Simplilearn. All rights reserved.

Docker Hub Login

Docker login command is used to connect to Docker hub from Docker client CLI. Before this can be done, you must tag existing image with Docker image concatenated with Docker username.

docker tag <image_name><tag> <docker_hub_username>/ <image_name><tag>
docker push <docker_hub_username>/ <image_name><tag>

```
root@docker:~# docker tag docker_apache anujsharma1990/docker_apache
root@docker:~# docker login
Login with your Docker ID to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://hub.docker.com to create one.
Username: anujsharma1990
Password:
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
root@docker:~# docker push anujsharma1990/docker_apache
The push refers to repository [docker.io/anujsharma1990/docker_apache]
e7e6e790a930: Pushed
735b1c81b582: Pushed
76c033092e10: Mounted from library/ubuntu
2146d867acf3: Mounted from library/ubuntu
ae1f631f14b7: Mounted from library/ubuntu
102645f1cf72: Mounted from library/ubuntu
latest: digest: sha256:7d8cccf25ef3b9f1578f0efb2d26a0c67cacc7c168709c12310c0ad47ad821e9 size: 1776
root@docker:~#
```

©Simplilearn. All rights reserved.

Containerization with Docker

Docker Networking

©Simplilearn. All rights reserved.

Docker Networking

Docker Networking helps Docker container to interact with other microservices over the network. Access to multiple applications hosted inside the Docker container is possible through Docker Networking.

There are four types of Docker networks:

- Bridge Mode: It is the default network available for Docker containers. It helps to access Docker container using same Docker host IP address and port.

©Simplilearn. All rights reserved.

Docker Networking (Contd.)

- Host Mode: This will inherit all network resources directly from host and can become exposed directly to public network. This is faster than Bridge mode, but there might be some security issues.

- Container Mode: This mode helps to have a unique IP address for each Docker container. You can not access the application through external server, but you can access it using Docker container IP through internal server.

- None: in this mode, networking gets switched off and Docker containers could not be able to connect to the network anymore. This helps to setup the applications network is not needed.

©Simplilearn. All rights reserved.

Execution on Docker Networking

You can use commands below to host an application in Docker container and also access it using a specific port from an external server. You can easily access application from anywhere through the internet.

```
root@docker:~# docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES
root@docker:~# docker run -d -p 8080:8080 --name jenkins jenkins
092942ab1619b3c853ed2c701b6060750c624562a0f3e9f4b2cb2607ab88f35d
root@docker:~# curl localhost:8080
<html><head><meta http-equiv='refresh' content='1;url=/login?from=%2F'/><script>window.location.replace('/login?from=%2F');</script></head><body style='bac
kground-color:white; color:white;'>


Authentication required
<!--
You are authenticated as: anonymous
Groups that you are in:

Permission you need to have (but didn't): hudson.model.Hudson.Administer
-->

</body></html>

    root@docker:~#
```

©Simplilearn. All rights reserved.

Assisted Practice

Duration: 45 mins

Demonstrate Docker Networking with Two SSHs

Problem Statement: You are given a project to demonstrate Docker networking and its configuration in a container.

Access: Click on the **Labs** tab on the left side panel of the LMS. Copy or note the username and password that is generated. Click on the **Launch Lab** button. On the page that appears, enter the username and password in the respective fields, and click **Login**.

©Simplilearn. All rights reserved.

Assisted Practice: Guidelines to Demonstrate Docker Networking

1. Login to your Ubuntu Lab and open the terminal.
2. Create a Centos container and install the net tools.
3. Confirm the IP address, hostname, and exit the container.
4. Commit the container to an image.
5. Create a bridge network and find its IP range.
6. Connect the network from the second SSH window.
7. Confirm the IP addresses and hostnames.
8. Disconnect the connection and exit the container.

©Simplilearn. All rights reserved.

Key Takeaways

You are now able to:

- Describe the importance of Docker over Virtual Machines
- Demonstrate on Docker images, Containers, and Docker Registry
- Demonstrate on Docker Compose and Docker Hub
- Describe the importance of Docker Networking

©Simplilearn. All rights reserved.

Knowledge Check

©Simplilearn. All rights reserved.

Knowledge Check 1	**Which one of the following platforms is supported by Docker?**

a. Linux

b. Windows

c. Mac OS

d. All of the Above

©Simplilearn. All rights reserved.

Knowledge Check 1	**Which one of the following platforms is supported by Docker?**

a. Linux

b. Windows

c. Mac OS

d. All of the Above

The correct answer is **d. All of the Above**

Docker is supported by all major platforms, including Linux, windows, and MAC OS. You can download and install Docker on any of these platforms.

©Simplilearn. All rights reserved.

Knowledge Check 2	**Docker containers running on the same host share operating system kernel, and this makes them fast.**

a. True

b. False

©Simplilearn. All rights reserved.

Knowledge Check 2	**Docker containers running on the same host share operating system kernel, and this makes them fast.**

a. True

b. False

The correct answer is **a. True**

Docker uses host machine kernel; this helps containers to load faster and effectively.

©Simplilearn. All rights reserved.

Knowledge Check 3	**Which one of below is a tool used for defining and creating multiple containers?**

a. Docker Swarm

b. Docker Hub

c. Docker Cloud

d. Docker Compose

©Simplilearn. All rights reserved.

Knowledge Check 3	**Which one of below is a tool used for defining and creating multiple containers?**

a. Docker Swarm

b. Docker Hub

c. Docker Cloud

d. Docker Compose

The correct answer is **d. Docker Compose**

Docker compose is the tool that can be used to define multiple containers; the containers can run using Docker compose command.

©Simplilearn. All rights reserved.

Knowledge Check 4	**Which of the following is a Cloud Service provided by AWS for hosting Docker images?**

a. Docker Hub

b. AWS ECR

c. Google Container Registry

d. Jfrog Artifactory

©Simplilearn. All rights reserved.

Knowledge Check

4

Which of the following is a Cloud Service provided by AWS for hosting Docker images?

a. Docker Hub

b. AWS ECR

c. Google Container Registry

d. Jfrog Artifactory

The correct answer is **b. AWS ECR**

Amazon ECR service is used to create private repositories; it will help you host your application in Docker images easily.

©Simplilearn. All rights reserved.

Lesson-End Project

Build a Docker Image and Deploy to the Docker Hub

Duration: 40 mins

Problem Statement:

Perform the following actions:

- Create a GitHub repository to save the code and to build the image.
- Create an account on Docker Hub.
- Install Docker on your machine.
- Login to Docker Hub.

Access: Click on the **Labs** tab on the left side panel of the LMS. Copy or note the username and password that is generated. Click on the **Launch Lab** button. On the page that appears, enter the username and password in the respective fields, and click **Login**.

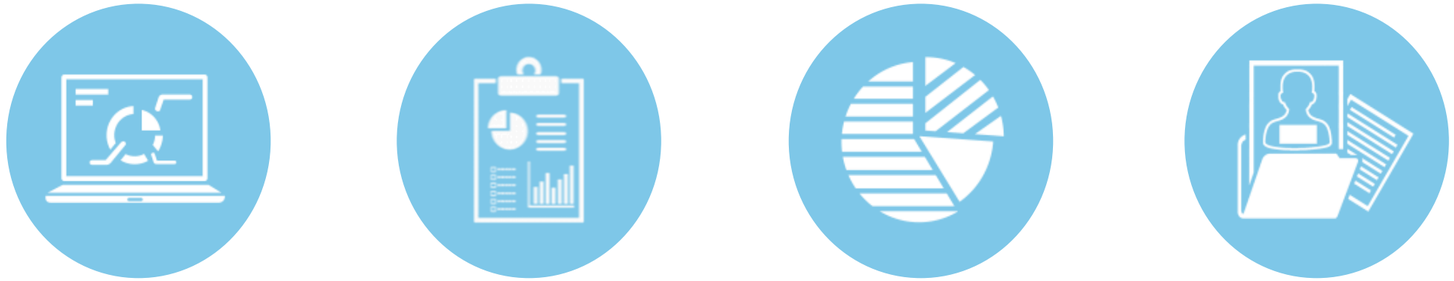
©Simplilearn. All rights reserved.

Thank You

©Simplilearn. All rights reserved.

DevOps Certification Training

Lesson 07: Continuous Monitoring



© Simplilearn. All rights reserved.

Learning Objectives

By the end of this lesson, you will be able to:

- Explain the role of continuous monitoring tools in DevOps
- Demonstrate Nagios
- Describe Grafana
- Describe ELK Stack
- Identify the suitable continuous monitoring tool for your organization

©Simplilearn. All rights reserved.

Continuous Monitoring

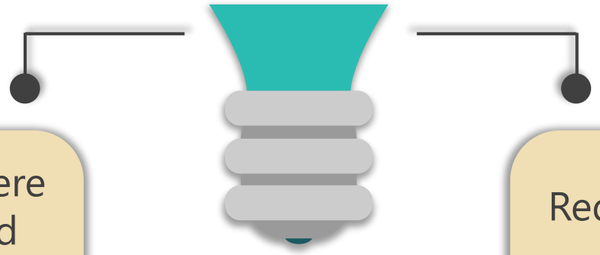
Overview of Continuous Monitoring

©Simplilearn. All rights reserved.

Continuous Monitoring

Helps detect risk, compliance, and security issues for an operational environment

Helps in continuous monitoring of environment so that adequate precautions are taken

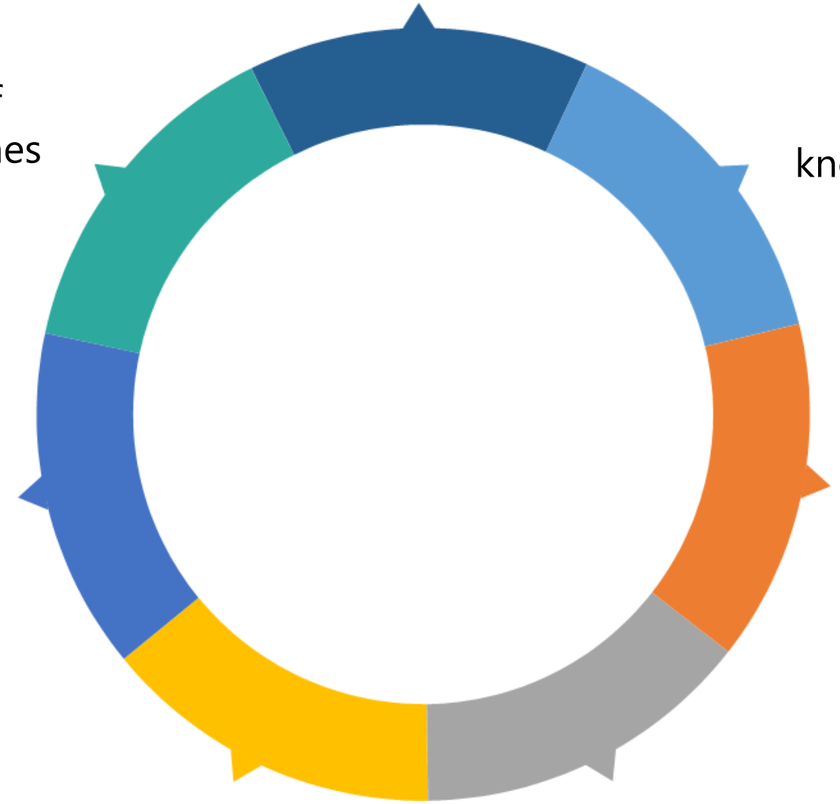


Acts as an auditing tool where it can navigate through old monitoring data to analyze and improve performance of system

Reduces the chances of failure and provides comfort to business in terms of security and reliability

©Simplilearn. All rights reserved.

Role of Monitoring Systems



©Simplilearn. All rights reserved.

Continuous Monitoring

Types of Monitoring System

©Simplilearn. All rights reserved.

Types of Monitoring

Real-Time Monitoring

- Server CPU stats
- Disk usage and memory stats
- Spikes in CPU performance
- I/O count on server

Infrastructure Monitoring

- CPU and memory
- Network and routers
- App servers, web servers, and DB servers

Application Monitoring

- API success/failures count
- API accessibility
- API HTTP error code

©Simplilearn. All rights reserved.

Popular Monitoring Tools

elastic + logstash + kibana

DATADOG

New Relic.

Ganglia

Nagios®

amazon web services™ CloudWatch

splunk>

pagerduty

Grafana

APPDYNAMICS

sensu

ZABBIX

©Simplilearn. All rights reserved.

DevOps Monitoring Tools

Nagios Core is a free open-source, application and infrastructure monitoring tool. Nagios was launched in 2002 and it became one of the popular monitoring tools in many organizations. It can monitor applications, networks, routers, switches, and servers. It needs Nagios NRPE agents to be deployed on respective servers to collect stats from node machines. Nagios enterprise version is also available.

ELK is a log monitoring and open-source tool. ELK is a combination of three open-source tools: Elasticsearch, Logstash, and Kibana. Elasticsearch is the heart of the stack since it acts as data engine, stores all applications, server logs, and fetches the data to analyze. Logstash acts as data pipeline which processes logs and helps in saving the data to Elasticsearch. Kibana is a front-end application used to visualize and display the data fetched from data engine.

©Simplilearn. All rights reserved.

DevOps Monitoring Tools (Contd.)

ZABBIX

Zabbix was launched in 2001 and is open-source toll that provides similar features like Nagios. It needs agents to be installed on the nodes in order to monitor the data.

sensu

Sensu is a powerful next-generation monitoring tool which is more popular compared to traditional monitoring tools. It was launched in 2011 as open-source under MIT license. Sensu enterprise version is available with additional features and plugins. It uses RabbitMQ to exchange data between nodes and master server. It uses Redis as datastore to store all the monitoring data.

©Simplilearn. All rights reserved.

DevOps Monitoring Tools (Contd.)

New Relic was launched in 2008 as SAAS(Software A As Service) software offering. It helps to monitor applications, and servers in real-time. New Relic's collectors installation in the nodes is necessary instead of New Relic software in the infrastructure. All monitoring data is transferred to New Relic and its dashboards are used to visualize monitoring data.

Splunk is interpreted as an application and security analytics tool. It collects data from each application and server and can be further analyzed to predict the future behavior for necessary precautions. Monitoring application failures and warning exceptions are possible. It is implemented in financial and product-based organizations to monitor the applications.

©Simplilearn. All rights reserved.

DevOps Monitoring Tools (Contd.)

DATADOG

Datadog is a cloud-based monitoring service. Datadog agent should be installed on the servers to monitor other servers within the infrastructure. All monitoring data is pushed to Datadog web application to visualize it.

APPDYNAMICS

AppDynamics tool is used to monitor the server and application performance which results in improved efficiency of the source code. It helps in making a suitable business decision while monitoring application, as it monitors both mobile and web.

©Simplilearn. All rights reserved.

DevOps Monitoring Tools (Contd.)

AWS CloudWatch is one of the core services of AWS cloud. By default, all the services in AWS are monitored by CloudWatch. It can store logs from various serverless components in AWS. It retains and stores monitored data, which is helpful to validate the stats anytime. It helps to create and generate alerts to users in case of issues.

©Simplilearn. All rights reserved.

Continuous Monitoring

Demonstrate Nagios

©Simplilearn. All rights reserved.

Nagios Installation

```
root@ip-172-31-15-211:~# curl https://assets.nagios.com/downloads/nagiosxi/install.sh | sh
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   629  100   629    0     0    436      0  0:00:01  0:00:01 --:--:--   435
/usr/bin/wget
Reading package lists... Done
Building dependency tree
Reading state information... Done
wget is already the newest version (1.19.4-1ubuntu2.1).
0 upgraded, 0 newly installed, 0 to remove and 105 not upgraded.
check if /tmp/nagiosxi exists
Downloading latest Nagios XI release
--2018-11-22 02:41:48--  https://assets.nagios.com/downloads/nagiosxi/xi-latest.tar.gz
Resolving assets.nagios.com (assets.nagios.com)... 72.14.181.71, 2600:3c00::f03c:91ff:fedf:b821
Connecting to assets.nagios.com (assets.nagios.com)|72.14.181.71|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 54397198 (52M) [application/x-gzip]
Saving to: â/tmp/xi-latest.tar.gzâ

/tmp/xi-latest.tar.gz                     100%[=================================================================

2018-11-22 02:42:07 (2.81 MB/s) - â/tmp/xi-latest.tar.gzâ saved [54397198/54397198]

Checking MySQL credentials...
MySQL not yet installed - that's okay.
Running './0-repos'...
Configuring Repos...
Repos configured OK
RESULT=0
Running './1-prereqs'...
Installing prerequisites...
Checking conflicting packages
```

©Simplilearn. All rights reserved.

Nagios Installation (Contd.)

```
root@ip-172-31-15-211:~# service postgresql status
â postgresql.service - PostgreSQL RDBMS
   Loaded: loaded (/lib/systemd/system/postgresql.service; enabled; vendor preset: enabled)
   Active: active (exited) since Thu 2018-11-22 02:46:47 UTC; 37min ago
 Main PID: 7625 (code=exited, status=0/SUCCESS)
    Tasks: 0 (limit: 1152)
   CGroup: /system.slice/postgresql.service

Nov 22 02:46:47 ip-172-31-15-211 systemd[1]: Starting PostgreSQL RDBMS...
Nov 22 02:46:47 ip-172-31-15-211 systemd[1]: Started PostgreSQL RDBMS.
root@ip-172-31-15-211:~# service apache2 status
â apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
  Drop-In: /lib/systemd/system/apache2.service.d
           ââapache2-systemd.conf
   Active: active (running) since Thu 2018-11-22 02:56:48 UTC; 28min ago
  Process: 4267 ExecStop=/usr/sbin/apachectl stop (code=exited, status=0/SUCCESS)
  Process: 11567 ExecReload=/usr/sbin/apachectl graceful (code=exited, status=0/SUCCESS)
  Process: 4272 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
 Main PID: 4287 (apache2)
    Tasks: 6 (limit: 1152)
   CGroup: /system.slice/apache2.service
           ââ 4287 /usr/sbin/apache2 -k start
           ââ11576 /usr/sbin/apache2 -k start
           ââ11577 /usr/sbin/apache2 -k start
           ââ11578 /usr/sbin/apache2 -k start
           ââ11579 /usr/sbin/apache2 -k start
           ââ11580 /usr/sbin/apache2 -k start

Nov 22 02:56:48 ip-172-31-15-211 systemd[1]: Stopped The Apache HTTP Server.
Nov 22 02:56:48 ip-172-31-15-211 systemd[1]: Starting The Apache HTTP Server...
Nov 22 02:56:48 ip-172-31-15-211 systemd[1]: Started The Apache HTTP Server.
Nov 22 03:09:11 ip-172-31-15-211 systemd[1]: Reloading The Apache HTTP Server.
Nov 22 03:09:12 ip-172-31-15-211 systemd[1]: Reloaded The Apache HTTP Server.
root@ip-172-31-15-211:~#
```

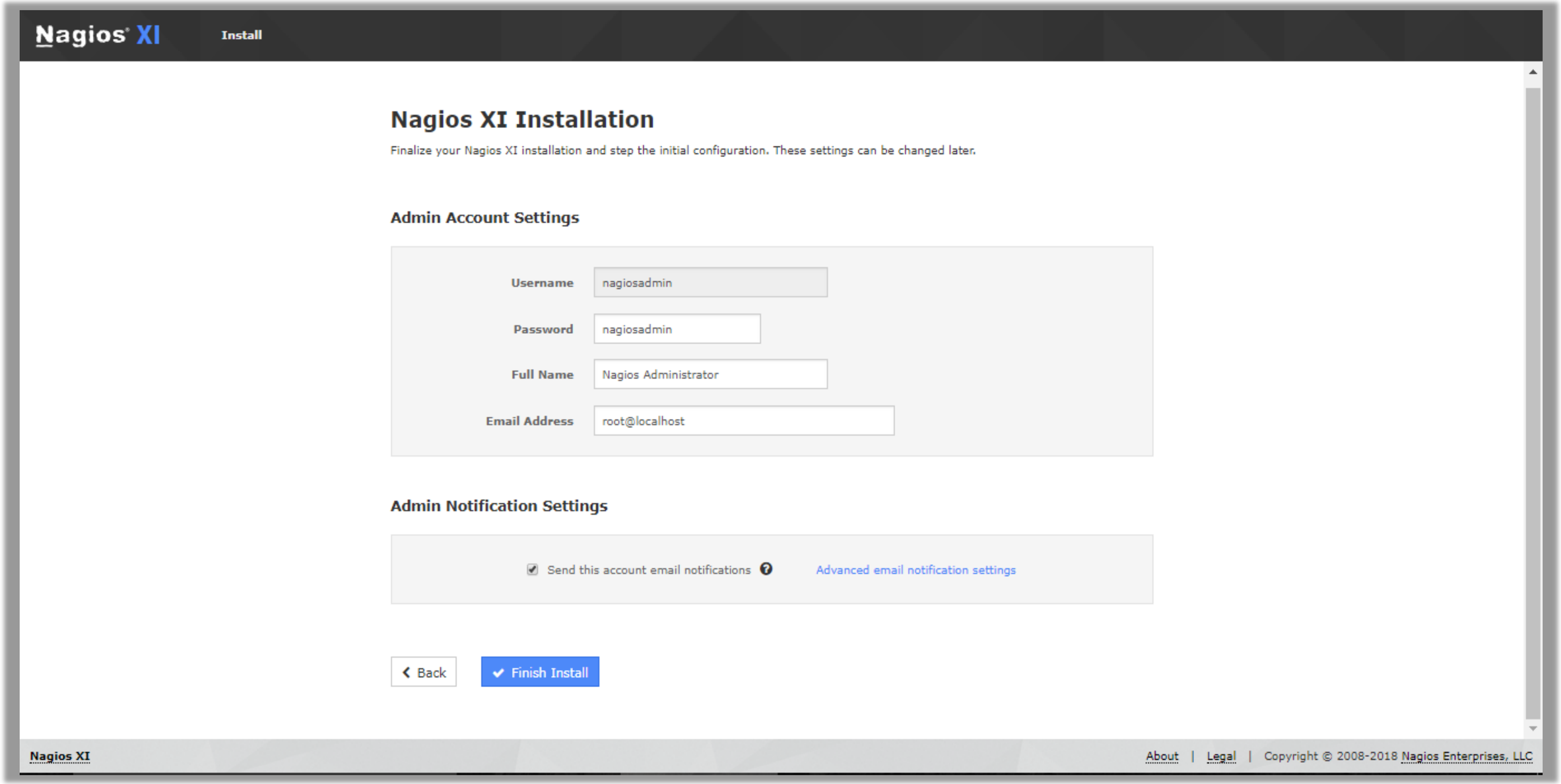
©Simplilearn. All rights reserved.

Nagios Installation (Contd.)

```
root@ip-172-31-15-211:~# service shellinabox status
â shellinabox.service - LSB: Shell In A Box Daemon
   Loaded: loaded (/etc/init.d/shellinabox; generated)
   Active: active (running) since Thu 2018-11-22 02:56:24 UTC; 32min ago
     Docs: man:systemd-sysv-generator(8)
  Process: 2955 ExecStop=/etc/init.d/shellinabox stop (code=exited, status=0/SUCCESS)
  Process: 2962 ExecStart=/etc/init.d/shellinabox start (code=exited, status=0/SUCCESS)
    Tasks: 2 (limit: 1152)
   CGroup: /system.slice/shellinabox.service
           ââ2997 /usr/bin/shellinaboxd -q --background=/var/run/shellinaboxd.pid -c /var/lib/shellinabox -p 7878 -u she
           ââ3003 /usr/bin/shellinaboxd -q --background=/var/run/shellinaboxd.pid -c /var/lib/shellinabox -p 7878 -u she

Nov 22 02:56:24 ip-172-31-15-211 systemd[1]: Stopped LSB: Shell In A Box Daemon.
Nov 22 02:56:24 ip-172-31-15-211 systemd[1]: Starting LSB: Shell In A Box Daemon...
Nov 22 02:56:24 ip-172-31-15-211 systemd[1]: Started LSB: Shell In A Box Daemon.
root@ip-172-31-15-211:~# service mysql status
â mysql.service - MySQL Community Server
   Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2018-11-22 02:52:51 UTC; 35min ago
 Main PID: 32491 (mysqld)
    Tasks: 30 (limit: 1152)
   CGroup: /system.slice/mysql.service
           ââ32491 /usr/sbin/mysqld --daemonize --pid-file=/run/mysqld/mysqld.pid

Nov 22 02:52:51 ip-172-31-15-211 systemd[1]: Stopped MySQL Community Server.
Nov 22 02:52:51 ip-172-31-15-211 systemd[1]: Starting MySQL Community Server...
Nov 22 02:52:51 ip-172-31-15-211 systemd[1]: Started MySQL Community Server.
root@ip-172-31-15-211:~#
```

©Simplilearn. All rights reserved.

Nagios: Account Setup



©Simplilearn. All rights reserved.

Nagios: Dashboard

Nagios XI | Home | Views | Dashboards | Reports | Configure | Tools | Help | Admin | nagiosadmin | Logout

Notice: This trial copy of Nagios XI will expire in 60 days. **Purchase a License Now** or **Enter your license key.**

Quick View

- Home Dashboard
- Tactical Overview
- Birdseye
- Operations Center
- Operations Screen
- Open Service Problems
- Open Host Problems
- All Service Problems
- All Host Problems
- Network Outages

Details

- Service Status
- Host Status
- Hostgroup Summary
- Hostgroup Overview
- Hostgroup Grid
- Servicegroup Summary
- Servicegroup Overview
- Servicegroup Grid
- BPI
- Metrics

Graphs

- Performance Graphs
- Graph Explorer

Maps

- BBmap
- Hypermap
- Minemap
- NagVis
- Network Status Map

Home Dashboard

Getting Started Guide

Common Tasks:

- Change your account settings
  Change your account password and general preferences.
- Change your notifications settings
  Change how and when you receive alert notifications.
- Configure your monitoring setup
  Add or modify items to be monitored with easy-to-use wizards.

Getting Started:

- **Learn about XI**
  Learn more about XI and its capabilities.
- **Signup for XI news**
  Stay informed on the latest updates and happenings for XI.

Administrative Tasks

Task
Initial Setup Tasks:

Host Status Summary

Up	Down	Unreachable	Pending
1	0	0	0
Unhandled		**Problems**	**All**
0		0	1

Last Updated: 2018-11-22 04:06:33

Service Status Summary

Ok	Warning	Unknown	Critical	Pending
8	0	0	0	4
Unhandled		**Problems**		**All**
0		0		12

Last Updated: 2018-11-22 04:06:33

We're Here To Help!

Our knowledgeable techs are happy to help you with any questions or problems you may have getting Nagios up and running.

- Support Forum / Customer Support Forum
- Help Resources
- Customer Ticket Support Center
- Customer Phone Support: +1 651-204-9102 Ext. 4

Start Monitoring

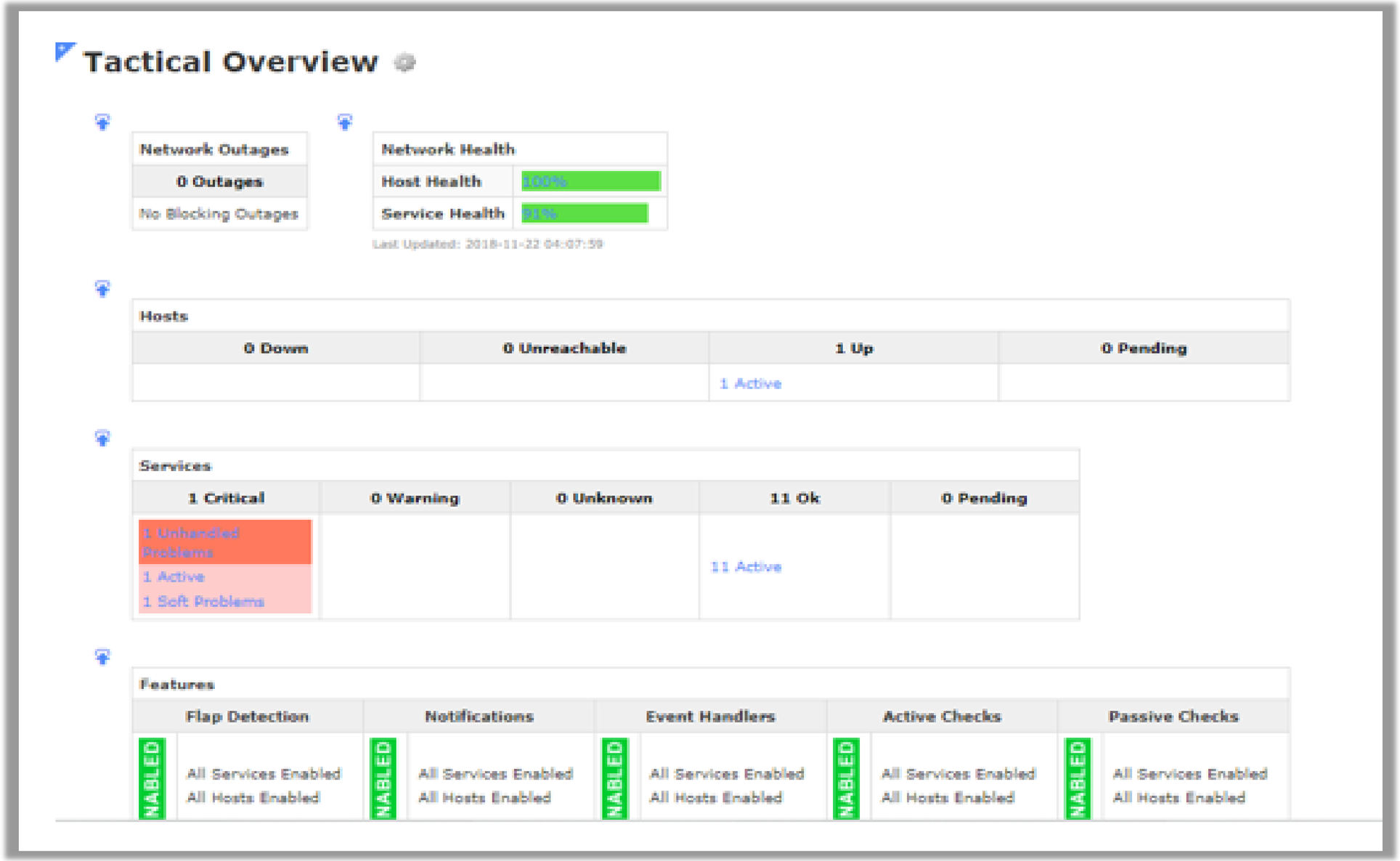
- Run a Config Wizard
- Run Auto-Discovery

Nagios XI 5.5.7 • Check for Updates

About | Legal | Copyright © 2008-2018 Nagios Enterprises, LLC

©Simplilearn. All rights reserved.

Nagios: Tactical Overview



©Simplilearn. All rights reserved.

Nagios: Service Status

Service Status

All services

Host Status Summary

Up	Down	Unreachable	Pending
1	0	0	0
Unhandled	**Problems**	**All**	
0	0	1	

Last Updated: 2018-11-22 04:08:42

Service Status Summary

Ok	Warning	Unknown	Critical	Pending
11	0	0	1	0
Unhandled		**Problems**		**All**
1		1		12

Last Updated: 2018-11-22 04:08:43

Search...

Showing 1-12 of 12 total records

Page 1 of 1 15 Per Page Go

Host	Service	Status	Duration	Attempt	Last Check	Status Information
localhost	Current Load	Ok	5m 51s	1/4	2018-11-22 04:08:22	OK - load average: 0.00, 0.29, 0.47
	Current Users	Ok	5m 26s	1/4	2018-11-22 04:08:47	USERS OK - 1 users currently logged in
	HTTP	Ok	5m 1s	1/4	2018-11-22 04:09:08	HTTP OK: HTTP/1.1 200 OK - 3160 bytes in 0.002 second response time
	PING	Ok	N/A	1/4	2018-11-22 04:04:37	PING OK - Packet loss = 0%, RTA = 0.04 ms
	Root Partition	Ok	N/A	1/4	2018-11-22 04:05:02	DISK OK - free space: / 4233 MB (53.85% inode=86%):
	Service Status - crond	Ok	N/A	1/4	2018-11-22 04:05:52	• cron.service - Regular background program processing daemon
	Service Status - httpd	Ok	N/A	1/4	2018-11-22 04:06:17	• apache2.service - The Apache HTTP Server
	Service Status - mysqld	Ok	N/A	1/4	2018-11-22 04:06:42	• mysql.service - MySQL Community Server
	Service Status - ndo2db	Ok	N/A	1/4	2018-11-22 04:07:07	• ndo2db.service - Nagios Data Out Daemon
	SSH	Ok	N/A	1/4	2018-11-22 04:05:27	SSH OK - OpenSSH_7.6p1 Ubuntu-4ubuntu0.1 (protocol 2.0)
	Swap Usage	Critical	1m 35s	2/4	2018-11-22 04:08:38	SWAP CRITICAL - 0% free (0 MB out of 0 MB) - Swap is either disabled, not present, or of zero size.
	Total Processes	Ok	N/A	1/4	2018-11-22 04:07:53	PROCS OK: 57 processes with STATE = RSZDT

Last Updated: 2018-11-22 04:09:13

©Simplilearn. All rights reserved.

Nagios: Host Status

Host Status

All hosts

Host Status Summary

Up	Down	Unreachable	Pending
1	0	0	0
Unhandled	**Problems**	**All**	
0	0	1	

Last Updated: 2018-11-22 04:11:04

Service Status Summary

Ok	Warning	Unknown	Critical	Pending
11	0	0	1	0
Unhandled		**Problems**		**All**
1		1		12

Last Updated: 2018-11-22 04:11:04

Search...

Showing 1-1 of 1 total records

Page 1 of 1 15 Per Page Go

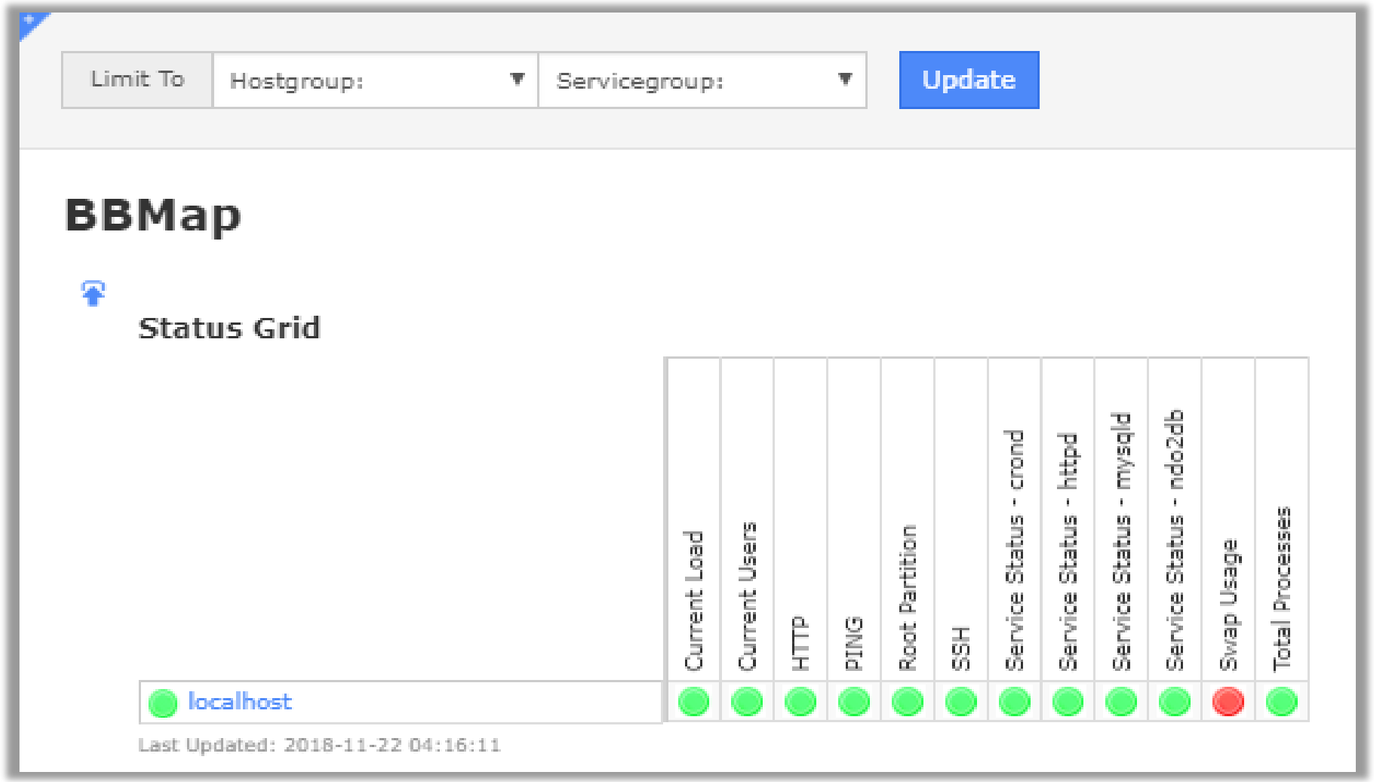
Host	Status	Duration	Attempt	Last Check	Status Information
localhost	Up	8m 6s	1/10	2018-11-22 04:07:38	OK - 127.0.0.1: rta 0.015ms, lost 0%

Last Updated: 2018-11-22 04:11:04

Page 1 of 1 15 Per Page Go

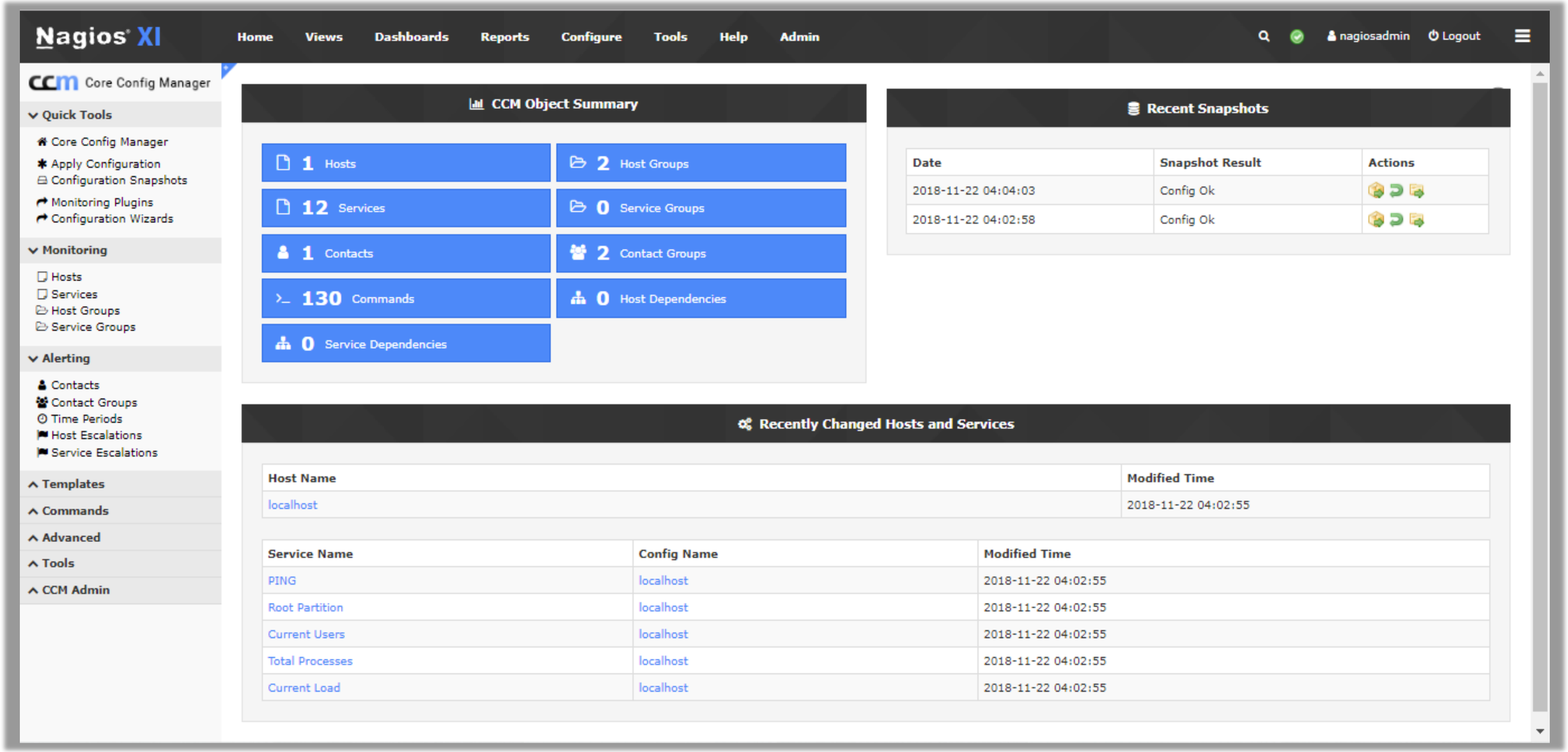
©Simplilearn. All rights reserved.

Nagios: Map Diagrams



©Simplilearn. All rights reserved.

Nagios: Core Configuration Manager



©Simplilearn. All rights reserved.

Assisted Practice

Duration: 90 mins

Working with Nagios Monitoring Tool

Problem Statement: You are given a project to demonstrate Nagios installation, install plugin, validate the installation, add a node, and validate the node details from Nagios dashboard.

Access: Click on the **Labs** tab on the left side panel of the LMS. Copy or note the username and password that is generated. Click on the **Launch Lab** button. On the page that appears, enter the username and password in the respective fields, and click **Login**.

©Simplilearn. All rights reserved.

Assisted Practice: Guidelines to Demonstrate Nagios Monitoring

1. Login to your Ubuntu Lab and open the terminal.
2. Download Nagios plugin source code.
3. Open Admin console of Nagios to install plugins.
4. Validate installation of Nagios plugin.
5. Add node details in Nagios portal.
6. Install Nagios agent on node machine.
7. Validate node details from Nagios dashboard.

©Simplilearn. All rights reserved.

Continuous Monitoring

Describe Grafana

©Simplilearn. All rights reserved.

Grafana Monitoring System

Grafana is an open-source monitoring system that supports alerts and graphical representation of monitoring stats from various sources.

Some of the companies using Grafana:

©Simplilearn. All rights reserved.

Features of Grafana Monitoring System

Visualize
Able to download Heatmaps, histograms, graphs, and geomaps

Alert
Defines thresholds visually and notifies via PagerDuty and Slack

Unify
Supports multiple databases which can be visualized in the Dashboard

Open-Source
Use Hosted Grafana or easily install on any platform.

Extend
The Official Grafana library has hundreds of dashboards and plugins to choose from

Collaborate
Shares data and dashboards across teams.

©Simplilearn. All rights reserved.

Grafana: Dashboard

©Simplilearn. All rights reserved.

Continuous Monitoring

Describe ELK Stack

©Simplilearn. All rights reserved.

ELK Stack

1. A set of utilities are available which combine together to provide the most powerful analytics for the business
2. Combination of Elasticsearch for searching data, Logstash to process and store various stats, and Kibana to visualize stats on front-end application
3. Open-source, collects logs from servers and applications that can be analyzed for improvement
4. Backup of monitoring stats and able to store our monitoring system
5. Browse through the stats to detect defects in application
6. Logstash can be used to gather stats from a variety of data sources and pushed to Kibana
7. Install clients to collect data and post it back to the ELK stack

©Simplilearn. All rights reserved.

ELK Stack Installation

```
root@ip-172-31-27-81:~# apt install apt-transport-https software-properties-common wget -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
software-properties-common is already the newest version (0.96.24.32.5).
wget is already the newest version (1.19.4-1ubuntu2.1).
apt-transport-https is already the newest version (1.6.6).
0 upgraded, 0 newly installed, 0 to remove and 105 not upgraded.
root@ip-172-31-27-81:~# apt -y install openjdk-8-jdk
Reading package lists... Done
Building dependency tree
Reading state information... Done
openjdk-8-jdk is already the newest version (8u181-b13-1ubuntu0.18.04.1).
0 upgraded, 0 newly installed, 0 to remove and 105 not upgraded.
root@ip-172-31-27-81:~# java -version
openjdk version "1.8.0_181"
OpenJDK Runtime Environment (build 1.8.0_181-8u181-b13-1ubuntu0.18.04.1-b13)
OpenJDK 64-Bit Server VM (build 25.181-b13, mixed mode)
root@ip-172-31-27-81:~#
```

©Simplilearn. All rights reserved.

ELK Stack Installation (Contd.)

```
root@ip-172-31-27-81:/etc/elasticsearch# wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add -
OK
root@ip-172-31-27-81:/etc/elasticsearch# echo "deb https://artifacts.elastic.co/packages/5.x/apt stable main" | sudo tee -a
deb https://artifacts.elastic.co/packages/5.x/apt stable main
root@ip-172-31-27-81:/etc/elasticsearch# vi /etc/apt/sources.list.d/elastic-5.x.list
root@ip-172-31-27-81:/etc/elasticsearch# apt install elasticsearch
Reading package lists... Done
Building dependency tree
Reading state information... Done
elasticsearch is already the newest version (5.6.13).
0 upgraded, 0 newly installed, 0 to remove and 105 not upgraded.
root@ip-172-31-27-81:/etc/elasticsearch# service elasticsearch restart
root@ip-172-31-27-81:/etc/elasticsearch# service elasticsearch status
â elasticsearch.service - Elasticsearch
   Loaded: loaded (/usr/lib/systemd/system/elasticsearch.service; disabled; vendor preset: enabled)
   Active: active (running) since Thu 2018-11-22 18:23:57 UTC; 3s ago
     Docs: http://www.elastic.co
  Process: 20146 ExecStartPre=/usr/share/elasticsearch/bin/elasticsearch-systemd-pre-exec (code=exited, status=0/SUCCESS)
 Main PID: 20155 (java)
    Tasks: 14 (limit: 1152)
   CGroup: /system.slice/elasticsearch.service
           ââ20155 /usr/bin/java -Xms300m -Xmx300m -XX:+UseConcMarkSweepGC -XX:CMSInitiatingOccupancyFraction=75 -XX:+UseCMS

Nov 22 18:23:57 ip-172-31-27-81 systemd[1]: Starting Elasticsearch...
Nov 22 18:23:57 ip-172-31-27-81 systemd[1]: Started Elasticsearch.
root@ip-172-31-27-81:/etc/elasticsearch#
```

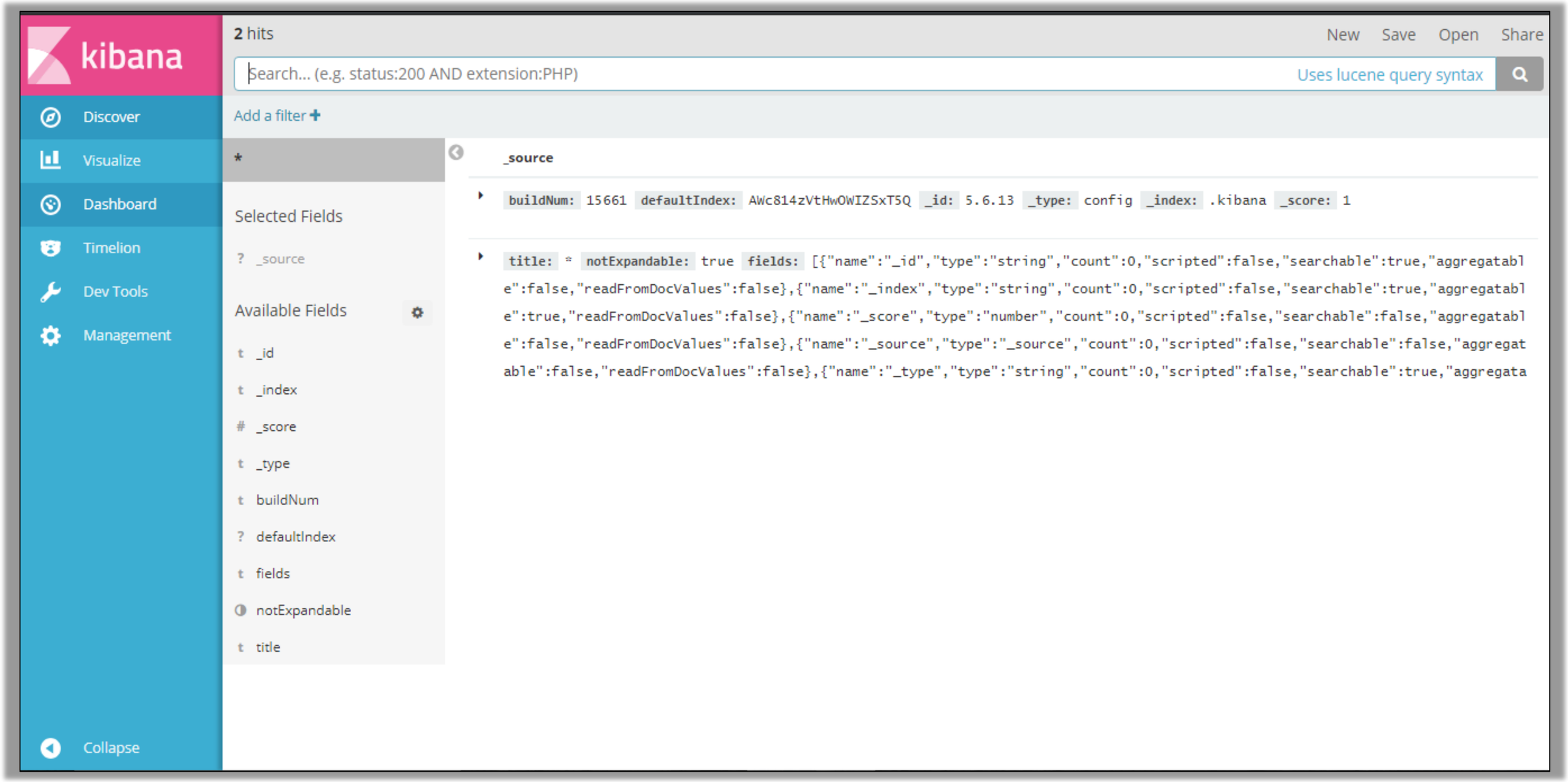
©Simplilearn. All rights reserved.

ELK Stack Installation (Contd.)

```
root@ip-172-31-27-81:/etc/elasticsearch# apt install kibana nginx
Reading package lists... Done
Building dependency tree
Reading state information... Done
nginx is already the newest version (1.14.0-0ubuntu1.2).
kibana is already the newest version (5.6.13).
0 upgraded, 0 newly installed, 0 to remove and 105 not upgraded.
root@ip-172-31-27-81:/etc/elasticsearch# echo "admin:$(openssl passwd -apr1 YourStrongPassword)" | sudo tee -a /etc/nginx/htpasswd.kibana
admin:$apr1$aWn15jgb$w52VfFuc67qTHfTd879gr0
root@ip-172-31-27-81:/etc/elasticsearch# service nginx restart
root@ip-172-31-27-81:/etc/elasticsearch# apt install logstash -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  logstash
0 upgraded, 1 newly installed, 0 to remove and 105 not upgraded.
Need to get 106 MB of archives.
After this operation, 201 MB of additional disk space will be used.
Get:1 https://artifacts.elastic.co/packages/5.x/apt stable/main amd64 logstash all 1:5.6.13-1 [106 MB]
Fetched 106 MB in 14s (7415 kB/s)
Selecting previously unselected package logstash.
(Reading database ... 112407 files and directories currently installed.)
Preparing to unpack .../logstash_1%3a5.6.13-1_all.deb ...
Unpacking logstash (1:5.6.13-1) ...
Setting up logstash (1:5.6.13-1) ...
Using provided startup.options file: /etc/logstash/startup.options
Successfully created system startup script for Logstash
root@ip-172-31-27-81:/etc/elasticsearch#
```

©Simplilearn. All rights reserved.

Kibana Visualizer Application



©Simplilearn. All rights reserved.

Key Takeaways

You are now able to:

- Explain the role of continuous monitoring tools in DevOps
- Demonstrate Nagios
- Describe Grafana
- Describe ELK Stack
- Identify the suitable continuous monitoring tool for your organization

©Simplilearn. All rights reserved.

Knowledge Check

©Simplilearn. All rights reserved.

Knowledge Check 1	**Which one of the following is a task of monitoring system?**

a. Server Monitoring

b. Application Monitoring

c. Network Monitoring

d. All of the above

©Simplilearn. All rights reserved.

Knowledge Check 1	**Which one of the following is a task of monitoring system?**

a. Server Monitoring

b. Application Monitoring

c. Network Monitoring

d. All of the above

The correct answer is **d. All of the above**

Server monitoring, application monitoring, and networking are used for monitoring stats of various infrastructure components.

©Simplilearn. All rights reserved.

Knowledge Check 2	**Which one of the following is the valid monitoring agent for Nagios in Linux environment?**

a. NRPE

b. NSClient ++

c. RDP

d. Agent is not required

©Simplilearn. All rights reserved.

Knowledge Check 2	**Which one of the following is the valid monitoring agent for Nagios in Linux environment?**

a. NRPE

b. NSClient ++

c. RDP

d. Agent is not required

The correct answer is **a. NRPE**

NRPE agent is installed on various Linux servers to gather various monitoring stats from server.

©Simplilearn. All rights reserved.

Knowledge Check 3	**Which one of the following is the data source in ELK Stack?**

a. Elasticsearch

b. Logstash

c. Kibana

d. None of above

©Simplilearn. All rights reserved.

Knowledge Check 3	**Which one of the following is the data source in ELK Stack?**

a. Elasticsearch

b. Logstash

c. Kibana

d. None of above

The correct answer is **a. Elasticsearch**

Elasticsearch is the data source used to store monitoring stats from various nodes.

©Simplilearn. All rights reserved.

Knowledge Check 4	**Which one of the following is the data pipeline component in ELK Stack?**

a. Elasticsearch

b. Logstash

c. Kibana

d. None of the above

©Simplilearn. All rights reserved.

Knowledge Check

4

Which one of the following is the data pipeline component in ELK Stack?

a. Elasticsearch

b. Logstash

c. Kibana

d. None of the above

The correct answer is **b. Logstash**

Logstash is the data pipeline component which is used to redirect the logs from nodes to datastore.

©Simplilearn. All rights reserved.

Knowledge Check 5	**Which one of the following is the web component in ELK Stack?**

a. Elasticsearch

b. Logstash

c. Kibana

d. None of the above

©Simplilearn. All rights reserved.

Knowledge Check 5	**Which one of the following is the web component in ELK Stack?**

a. Elasticsearch

b. Logstash

c. Kibana

d. None of the above

The correct answer is **c. Kibana**

Kibana is the component used to visualize monitoring stats fetched from Elasticsearch.

©Simplilearn. All rights reserved.

Lesson-End Project

Add a Node in Nagios Monitoring Tool

Duration: 60 mins

Problem Statement:

Perform the following actions:

- Download Nagios plugin source code.
- Open Admin console of Nagios to install plugin.
- Find the required plugin.
- Install the plugin.
- Validate if it is working.

Access: Click on the **Labs** tab on the left side panel of the LMS. Copy or note the username and password that is generated. Click on the **Launch Lab** button. On the page that appears, enter the username and password in the respective fields, and click **Login**.

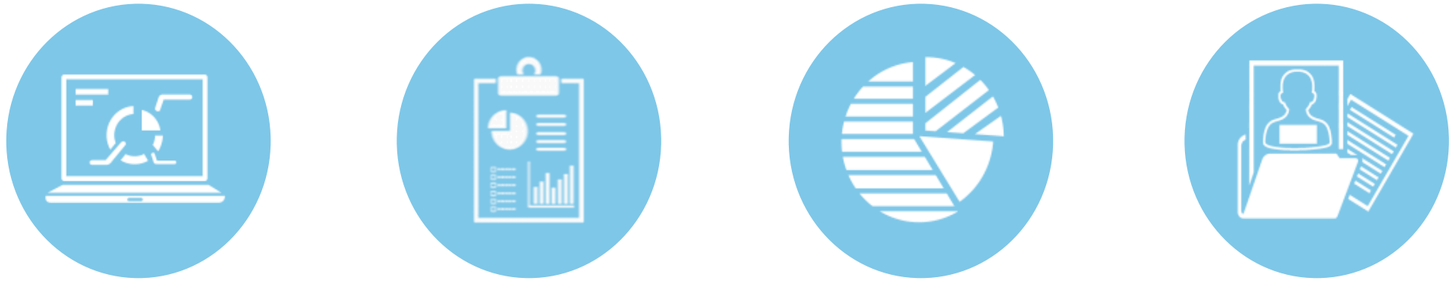
©Simplilearn. All rights reserved.

Thank You

©Simplilearn. All rights reserved.

DevOps Certification Training

Lesson 08: Need of Cloud in DevOps



© Simplilearn. All rights reserved.

Learning Objectives

By the end of this lesson, you will be able to:

- Describe the concepts of cloud computing
- Explain the importance of cloud in DevOps
- Explain the need of AWS in DevOps
- Demonstrate the use of Kubernetes

©Simplilearn. All rights reserved.

Need of Cloud in DevOps

Overview of Cloud Computing

©Simplilearn. All rights reserved.

Cloud Computing

Parallel and distributed systems manages through the internet

Manages virtual resources with minimum maintenance

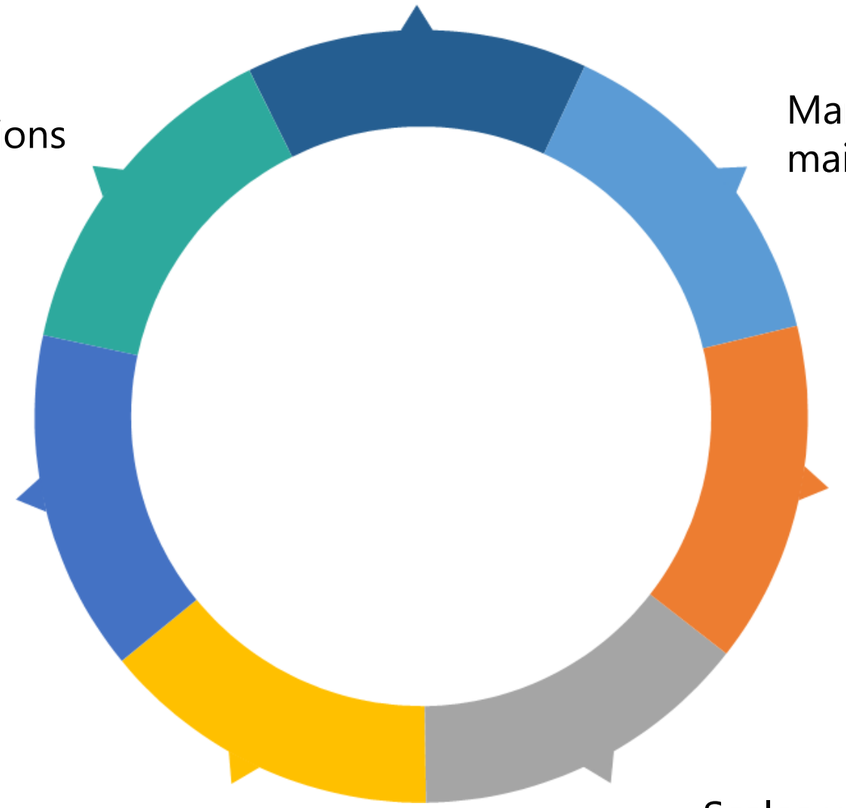
Shared pool of computing resources interconnected over networks

Scales up and down resources through the internet

Easy to implement and use

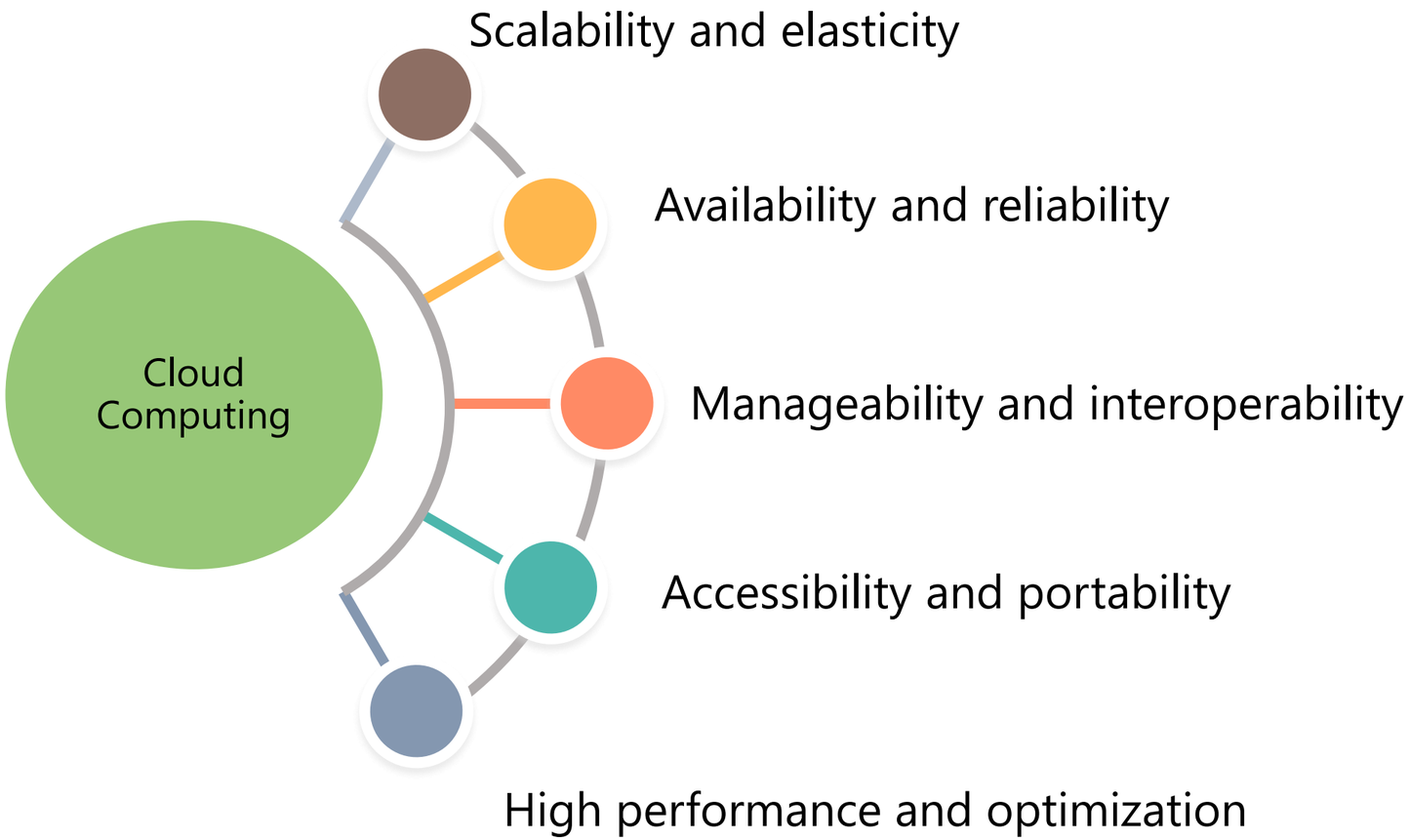
On-demand self service portal used to facilitate computing resources

Fast and inexpensive computing solutions



©Simplilearn. All rights reserved.

Cloud Computing Characteristics



©Simplilearn. All rights reserved.

Cloud Computing Characteristics (Contd.)

- **Scalability**: Ability to handle increasing work load and variations in computing power
- **Elasticity**: Ability to process workload by provisioning and deprovisioning of infrastructure
- **Availability:** Availability of the system in an operational state and representing uptime of the infrastructure
- **Reliability:** Ability to maintain stable functionality of the system over a period of time
- **Manageability**: Ability to manage various components in a cloud network

©Simplilearn. All rights reserved.

Cloud Computing Characteristics (Contd.)

- **Interoperability**: Ability to integrates the systems without any restrictions
- **Accessibility**: Ability to provide an environment that can be accessed by multiple users
- **Portability**: Ability to access any service and environment from any platform
- **Performance**: Ability to design a high performance environment
- **Optimization**: Ability to stabilize the cloud environment and its efficiency

©Simplilearn. All rights reserved.

Advantages of Cloud Computing

1. Helps to manage infrastructure even without any infrastructure knowledge
2. Provides updated applications and services. It increases the efficiency of applications hosted on cloud
3. Provides cost-effective measures to set up the architecture to use cloud resources
4. Follows a pay-per-use model in which users pay only for the time they use cloud infrastructure
5. Stores applications and data on servers which are easily accessible through internet
6. Helps users easily access the changes made to the cloud from any network or system
7. Provides faster server performance which makes it quick and easy to boot systems

©Simplilearn. All rights reserved.

Disadvantages of Cloud Computing

1. Needs continuous internet connection
2. Provides users no control over the way cloud resources are shared and used which results in security threats
3. Does not work well in slow internet connections
4. Has no custom backup available for the data stored on it's servers
5. Lacks certain features found in desktop software such as Excel and Google Sheets
6. Performs slow in shared pools of infrastructure environment

©Simplilearn. All rights reserved.

Need of Cloud in DevOps

Cloud Services and Models

©Simplilearn. All rights reserved.

Cloud Service Models

PaaS Model

- Platform as a Service (PaaS) offers various types of development environments which can be used for product development.
- Examples: Google App Engine, Azure, and AWS Elastic Beanstalk

SaaS Model

- Software as a Service (SaaS) is an online service in which application software is provided to multiple users at a specific price.
- Examples: Google Apps, Salesforce.com, and CRM

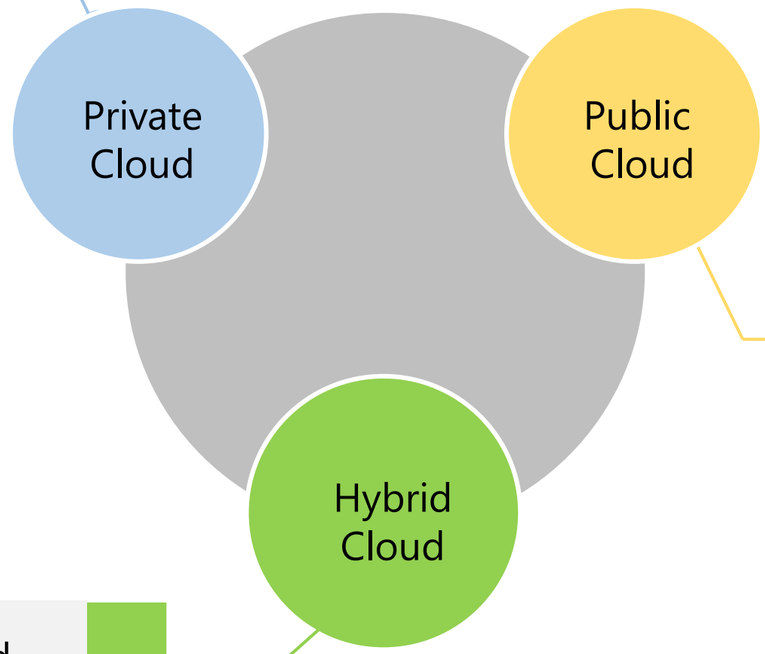
IaaS Model

- Infrastructure as a Service (IaaS) is an online service which provides physical resources like servers, databases, load balancers, and storage.
- Examples: EC2, virtual machines, compute engines, load balancers, and firewalls

©Simplilearn. All rights reserved.

Cloud Models

Cloud infrastructure deployed on the private servers of organizations. It is secure and reliable. Examples: Data Centers and OpenNebula.



Cloud service on a public cloud is less secure than on a private network. Examples:AWS, Azure, Google Cloud.

Combination of both, public and private cloud. It is a composition of different cloud providers.

©Simplilearn. All rights reserved.

Popular Cloud Providers

amazon web services

Microsoft Azure

Google Cloud Platform

IBM Cloud

Alibaba Cloud

DigitalOcean

©Simplilearn. All rights reserved.

Need of Cloud in DevOps

Using AWS in DevOps

©Simplilearn. All rights reserved.

simplilearn

CI/CD in AWS

Continuous integration is a process of merging all working copies of developers to main repositories on a daily basis. It deals with building source codes and executing the initial test cases.

Continuous deployment performs automatic deployment once the build is successful. This is the last step in designing a complete workflow of build and deployment automation. It helps in quicker development and deployment.

©Simplilearn. All rights reserved.

CI/CD Services in AWS

AWS CodeCommit

AWS CodeDeploy

AWS CodeBuild

AWS CodePipeline

©Simplilearn. All rights reserved.

CI/CD Services in AWS (Contd.)

AWS CodeCommit

CodeCommit is a highly scalable and secure source code management system. It provides private Git repositories. Uptime and space utilization are the major drawbacks. Multiple repositories can be created in it.

AWS CodeBuild

AWS CodeBuild helps to implement continuous integration for the cloud applications. You can compile source codes, execute test cases, and generate binaries from source codes for deployment. Private build servers and continuous integration environment will not be created. It can be used for multiple applications and various platforms.

©Simplilearn. All rights reserved.

CI/CD Services in AWS (Contd.)

AWS CodePipeline

CodePipeline is a continuous delivery service which helps to automate builds, tests, and deploy your application. It helps to release the versions of applications as source code changes. It is a single solution to performing a complete CI/CD workflow.

AWS CodeDeploy

CodeDeploy is a cloud deployment service which helps to perform deployment automation to EC2 servers, Lambda, Elastic Beanstalk, and other physical systems. It supports easy deployment without manual errors. It eliminates the additional step of configuring Jenkins or other deploy tools.

©Simplilearn. All rights reserved.

Need of Cloud in DevOps

Kubernetes

©Simplilearn. All rights reserved.

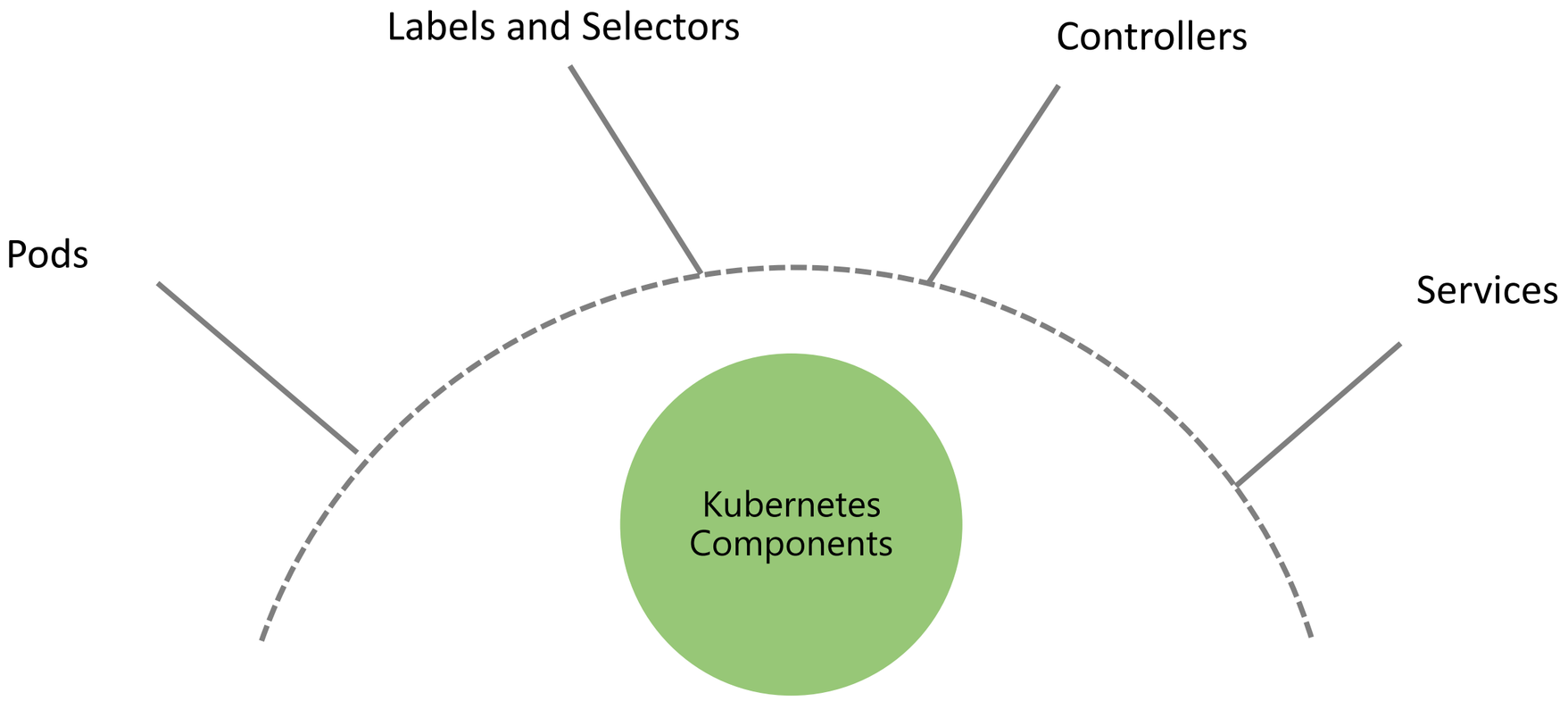
Kubernetes

- Open-source orchestration tool used to manage containerized applications
- Originally designed by Google and acquired by Cloud Native Computing Foundation
- Automates deployment and scales containers across multiple nodes in Kubernetes clusters
- Designed in GO language
- Acts as a cloud service in major cloud providers such as EKS in AWS and Kubernetes Engine in GCP

©Simplilearn. All rights reserved.

Kubernetes Components

Kubernetes is a combination of various building blocks which collectively help to manage, deploy, and scale containerized applications.



©Simplilearn. All rights reserved.

Pods

- Pod is a basic scheduling unit in Kubernetes
- Each pod comprises one or more containers that can be initialized on any host
- Each pod is assigned a unique IP using which we can redirect traffic from outside to the pod
- Pods are managed using the kubelet command line in a Kubernetes cluster
- Containers in a pod can consist of multiple applications
- Pod templates are used to define how pods will be created and deployed
- Pods share physical resources from host machines in forms of CPU, RAM, and storage

©Simplilearn. All rights reserved.

Labels and Selectors

- Kubernetes attaches key-value pairs called labels for various objects such as services, pods, and nodes
- These labels can be used to locate a specific resource
- Same label can be used for multiple objects, so you should define and create unique labels for Kubernetes objects

©Simplilearn. All rights reserved.

Controllers

- Controllers bring pods to a specific state
- ReplicationController replicates and scales pods across Kubernetes clusters
- Controllers take care of availability of pods, and if it fails, a replacement pod gets created automatically
- DaemonSet controller ensures only one pod runs on each node
- Job controller manages all the batch jobs of pods which are executed in a Kubernetes cluster
- Controllers manage pods using labels and selectors to identify resources

©Simplilearn. All rights reserved.

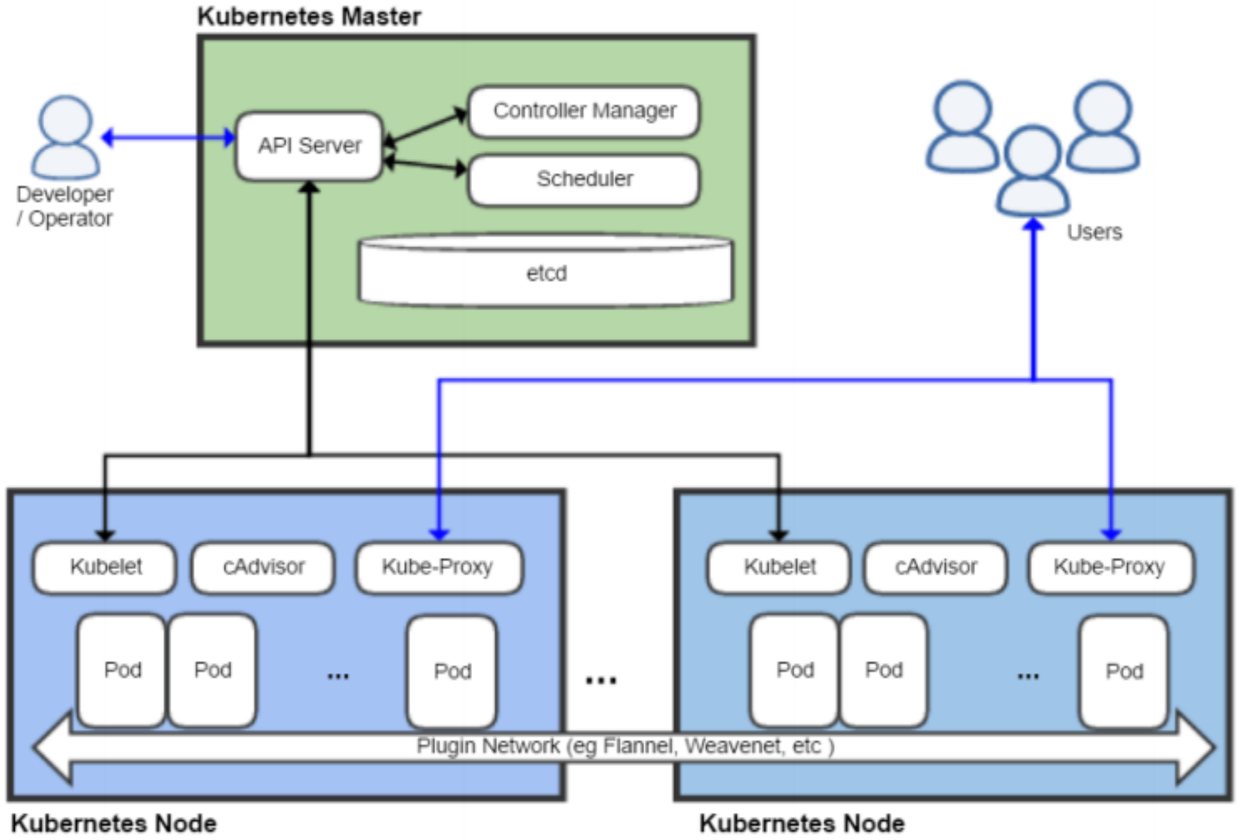
Services

- Collection of pods are bundled together in a service
- Kubernetes allocates a unique port and DNS to each service. The port and DNS details are changed only if the service object is recreated
- There can be multiple replicated pods in a service
- In case of multiple pods, an in-built load balancer is used to share the load between pods running on different nodes
- A service implements high availability and load share for containerized applications
- If a pod is terminated, a replacement pod will be initialized automatically

©Simplilearn. All rights reserved.

Kubernetes Architecture

- Uses master-slave architecture
- Kubernetes master contains the following architecture components:
  - etcd
  - API server
  - Scheduler
  - Controller manager
- Kubernetes client contains the following architecture components:
  - Kubelet
  - cAdvisor
  - Pod
  - Kube-Proxy



©Simplilearn. All rights reserved.

Kubernetes Master Components

etcd

etcd is a persistent, lightweight, and key-value data store. It stores the complete configuration data of a Kubernetes cluster. At any point of time you can check the state of a cluster with the available data. This data store can be shared with other components. It provides a data layer in Kubernetes clusters.

API Server

API Server supports Kubernetes API and processes all the requests from various components. It handles the REST requests and JSON requests and updates the state of each object in etcd.

©Simplilearn. All rights reserved.

Kubernetes Master Components (Contd.)

Scheduler

Scheduler is the component of Kubernetes responsible for managing workloads in a cluster. It identifies the unutilized node and the process to schedule pods on unutilized nodes based on the requirements. It helps to manage all Kubernetes resources effectively.

Controller Manager

Controller manager manages all controllers in Kubernetes such as DaemonSet and ReplicationController. It interacts with the API server to create, edit, and delete any resources being managed.

©Simplilearn. All rights reserved.

Kubernetes Node Components

kube-proxy

kube-proxy implements network proxy and acts as a load balancer in Kubernetes cluster. It helps to redirect traffic to a specific container in a pod based on the incoming port and IP details.

cAdvisor

cAdvisor is an agent that monitors and gathers resource usage and performance metrics such as CPU, memory, files, and network usage of containers on each node.

©Simplilearn. All rights reserved.

Kubernetes Node Components (Contd.)

Kubelet

Kubelet is responsible for the working of each node and ensuring the container's health. It monitors how the pods start, stop, and are maintained. Once the master detects a node failure, the ReplicationController observes the change in state and launches pods on other healthy nodes.

©Simplilearn. All rights reserved.

Need of Cloud in DevOps

Demonstration of the “kubectl” Command

©Simplilearn. All rights reserved.

Kubernetes Installation

```
root@docker:~# apt-get install -y curl apt-transport-https docker.io
Reading package lists... Done
Building dependency tree
Reading state information... Done
curl is already the newest version (7.58.0-2ubuntu3.5).
apt-transport-https is already the newest version (1.6.6).
docker.io is already the newest version (18.06.1-0ubuntu1~18.04.1).
0 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.
root@docker:~# curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | apt-key add -
OK
root@docker:~# echo "deb http://apt.kubernetes.io/ kubernetes-xenial main" >/etc/apt/sources.list.d/kubernet
root@docker:~# apt-get update
Hit:1 http://us-east1.gce.archive.ubuntu.com/ubuntu bionic InRelease
Hit:2 http://us-east1.gce.archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:3 http://us-east1.gce.archive.ubuntu.com/ubuntu bionic-backports InRelease
Hit:4 http://archive.canonical.com/ubuntu bionic InRelease
Get:5 https://packages.cloud.google.com/apt kubernetes-xenial InRelease [8993 B]
Ign:6 https://pkg.jenkins.io/debian-stable binary/ InRelease
Get:7 https://packages.cloud.google.com/apt kubernetes-xenial/main amd64 Packages [21.6 kB]
Hit:8 https://pkg.jenkins.io/debian-stable binary/ Release
Hit:9 http://security.ubuntu.com/ubuntu bionic-security InRelease
Fetched 30.6 kB in 1s (44.8 kB/s)
Reading package lists... Done
root@docker:~#
```

©Simplilearn. All rights reserved.

Kubernetes Installation (Contd.)

```
root@docker:~# apt-get install -y kubelet kubeadm kubectl
Reading package lists... Done
Building dependency tree
Reading state information... Done
kubeadm is already the newest version (1.12.3-00).
kubectl is already the newest version (1.12.3-00).
kubelet is already the newest version (1.12.3-00).
0 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.
root@docker:~# kubeadm init
[init] using Kubernetes version: v1.12.3
[preflight] running pre-flight checks
        [WARNING Service-Docker]: docker service is not enabled, please run 'systemctl enable docker.service'
[preflight/images] Pulling images required for setting up a Kubernetes cluster
[preflight/images] This might take a minute or two, depending on the speed of your internet connection
[preflight/images] You can also perform this action in beforehand using 'kubeadm config images pull'
[kubelet] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[preflight] Activating the kubelet service
[certificates] Generated ca certificate and key.
[certificates] Generated apiserver certificate and key.
[certificates] apiserver serving cert is signed for DNS names [docker kubernetes kubernetes.default kubernetes.de
 [10.96.0.1 10.142.0.4]
[certificates] Generated apiserver-kubelet-client certificate and key.
[certificates] Generated front-proxy-ca certificate and key.
[certificates] Generated front-proxy-client certificate and key.
[certificates] Generated etcd/ca certificate and key.
[certificates] Generated etcd/peer certificate and key.
```

©Simplilearn. All rights reserved.

Kubernetes Installation (Contd.)

```
[addons] Applied essential addon: CoreDNS
[addons] Applied essential addon: kube-proxy

Your Kubernetes master has initialized successfully!

To start using your cluster, you need to run the following as a regular user:

  mkdir -p $HOME/.kube
  sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
  sudo chown $(id -u):$(id -g) $HOME/.kube/config

You should now deploy a pod network to the cluster.
Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:
  https://kubernetes.io/docs/concepts/cluster-administration/addons/

You can now join any number of machines by running the following on each node
as root:

  kubeadm join 10.142.0.4:6443 --token irbw21.3po050fmlt0nqqu1 --discovery-token-ca-cert-hash sha256:c7e5ee2

root@docker:~#
```

©Simplilearn. All rights reserved.

Kubernetes Installation (Contd.)

```
root@docker:~# kubectl get node
NAME      STATUS   ROLES    AGE     VERSION
docker    Ready    master   5m46s   v1.12.3
root@docker:~# kubectl get pods --all-namespaces
NAMESPACE     NAME                                READY   STATUS    RESTARTS   AGE
kube-system   coredns-576cbf47c7-ggmhc            1/1     Running   0          5m45s
kube-system   coredns-576cbf47c7-xtxqj            1/1     Running   0          5m45s
kube-system   etcd-docker                         1/1     Running   0          5m1s
kube-system   kube-apiserver-docker               1/1     Running   0          4m55s
kube-system   kube-controller-manager-docker      1/1     Running   0          4m52s
kube-system   kube-proxy-r95g8                    1/1     Running   0          5m45s
kube-system   kube-scheduler-docker               1/1     Running   0          4m57s
kube-system   weave-net-bmhj6                     2/2     Running   0          31s
root@docker:~# kubectl create namespace application
namespace/application created
root@docker:~# kubectl get pods --all-namespaces
NAMESPACE     NAME                                READY   STATUS    RESTARTS   AGE
kube-system   coredns-576cbf47c7-ggmhc            1/1     Running   0          6m28s
kube-system   coredns-576cbf47c7-xtxqj            1/1     Running   0          6m28s
kube-system   etcd-docker                         1/1     Running   0          5m44s
kube-system   kube-apiserver-docker               1/1     Running   0          5m38s
kube-system   kube-controller-manager-docker      1/1     Running   0          5m35s
kube-system   kube-proxy-r95g8                    1/1     Running   0          6m28s
kube-system   kube-scheduler-docker               1/1     Running   0          5m40s
kube-system   weave-net-bmhj6                     2/2     Running   0          74s
```

©Simplilearn. All rights reserved.

Kubernetes Installation (Contd.)

```
root@docker:~# kubectl run kubernetes-bootcamp --image=docker.io/jocatalin/kubernetes-bootcamp:v1 --port=8080
kubectl run --generator=deployment/apps.v1beta1 is DEPRECATED and will be removed in a future version. Use kub
deployment.apps/kubernetes-bootcamp created
root@docker:~# kubectl get services
NAME         TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
kubernetes   ClusterIP   10.96.0.1    <none>        443/TCP   15m
root@docker:~# kubectl expose deployment/kubernetes-bootcamp --port=8080 --target-port=8080 --type=NodePort
service/kubernetes-bootcamp exposed
root@docker:~# kubectl describe services kubernetes-bootcamp | grep -i port
Type:                     NodePort
Port:                     <unset>  8080/TCP
TargetPort:               8080/TCP
NodePort:                 <unset>  31319/TCP
root@docker:~#  kubectl get pods
NAME                                  READY   STATUS    RESTARTS   AGE
kubernetes-bootcamp-7476558597-r5xw8   1/1     Running   0          30s
root@docker:~# kubectl exec -ti kubernetes-bootcamp-7476558597-r5xw8 curl localhost:8080
Hello Kubernetes bootcamp! | Running on: kubernetes-bootcamp-7476558597-r5xw8 | v=1
root@docker:~# curl localhost:31319
Hello Kubernetes bootcamp! | Running on: kubernetes-bootcamp-7476558597-r5xw8 | v=1
root@docker:~# kubectl get deployments
NAME                  DESIRED   CURRENT   UP-TO-DATE   AVAILABLE   AGE
kubernetes-bootcamp   1         1         1            1           81s
root@docker:~#
```

©Simplilearn. All rights reserved.

Kubernetes Installation (Contd.)

```
root@docker:~# kubectl scale deployments/kubernetes-bootcamp --replicas=2
deployment.extensions/kubernetes-bootcamp scaled
root@docker:~# kubectl get deployments
NAME                  DESIRED   CURRENT   UP-TO-DATE   AVAILABLE   AGE
kubernetes-bootcamp   2         2         2            2           2m34s
root@docker:~# kubectl get pods
NAME                                   READY   STATUS    RESTARTS   AGE
kubernetes-bootcamp-7476558597-kxp6z   1/1     Running   0          23s
kubernetes-bootcamp-7476558597-r5xw8   1/1     Running   0          2m37s
root@docker:~# curl localhost:31319
Hello Kubernetes bootcamp! | Running on: kubernetes-bootcamp-7476558597-kxp6z | v=1
root@docker:~# curl localhost:31319
Hello Kubernetes bootcamp! | Running on: kubernetes-bootcamp-7476558597-r5xw8 | v=1
root@docker:~# curl localhost:31319
Hello Kubernetes bootcamp! | Running on: kubernetes-bootcamp-7476558597-kxp6z | v=1
root@docker:~# curl localhost:31319
Hello Kubernetes bootcamp! | Running on: kubernetes-bootcamp-7476558597-r5xw8 | v=1
root@docker:~# curl localhost:31319
Hello Kubernetes bootcamp! | Running on: kubernetes-bootcamp-7476558597-r5xw8 | v=1
root@docker:~# kubectl describe services kubernetes-bootcamp | grep -i port
Type:                     NodePort
Port:                     <unset>  8080/TCP
TargetPort:               8080/TCP
NodePort:                 <unset>  31319/TCP
root@docker:~# curl localhost:31319
Hello Kubernetes bootcamp! | Running on: kubernetes-bootcamp-7476558597-r5xw8 | v=1
root@docker:~# kubectl get pods -o wide
NAME                                   READY   STATUS    RESTARTS   AGE     IP          NODE     NOMINATED NODE
kubernetes-bootcamp-7476558597-kxp6z   1/1     Running   0          2m5s    10.32.0.5   docker   <none>
kubernetes-bootcamp-7476558597-r5xw8   1/1     Running   0          4m19s   10.32.0.4   docker   <none>
```

©Simplilearn. All rights reserved.

Assisted Practice

Duration: 40 mins

Add a Linux Node to the Kubernetes Cluster

Problem Statement: You are given a project to demonstrate the installation of Kubernetes, configure pods, and add a node to the Kubernetes cluster.

Access: Click on the **Labs** tab on the left side panel of the LMS. Copy or note the username and password that is generated. Click on the **Launch Lab** button. On the page that appears, enter the username and password in their respective fields, and click **Login**.

©Simplilearn. All rights reserved.

Assisted Practice: Guidelines to the Demonstration of Adding a Node to the Cluster

1. Login to your Ubuntu Lab, and open the terminal.
2. Install Docker from the official site and check the version.
3. Enable Docker service to start on system boot.
4. Install Kubernetes, and update the apt-get package.
5. Install the tools: kubelet, kubeadm, and kubectl.
6. Apply the network configurations to the Kubernetes cluster.
7. Check the result of the pods.
8. Add a node to the existing Kubernetes cluster.

©Simplilearn. All rights reserved.

Key Takeaways

You are now able to:

- Describe the concepts of cloud computing
- Explain the importance of cloud in DevOps
- Explain the need of AWS in DevOps
- Demonstrate the use of Kubernetes

©Simplilearn. All rights reserved.

Knowledge Check

©Simplilearn. All rights reserved.

Knowledge Check 1	**Which one of the following cloud service models provides tools and environments to deploy applications?**

a. PaaS

b. SaaS

c. IaaS

d. All of the above

©Simplilearn. All rights reserved.

Knowledge Check 1	**Which one of the following cloud service models provides tools and environments to deploy applications?**

a. PaaS

b. SaaS

c. IaaS

d. All of the above

The correct answer is **a. PaaS**

PaaS provides the platform that can be used to host an application without any setup.

©Simplilearn. All rights reserved.

Knowledge Check 2	**Which of the following is the most important area of concern in cloud computing?**

a. Security

b. Storage

c. Scalability

d. All of the above

©Simplilearn. All rights reserved.

Knowledge Check 2	**Which of the following is the most important area of concern in cloud computing?**

a. Security

b. Storage

c. Scalability

d. All of the above

The correct answer is **a. Security**

Security is one of the areas of concern which can impact environment on a high scale.

©Simplilearn. All rights reserved.

Knowledge Check 3	**Which of the following will run on Kubernetes nodes?**

a. Kubelet

b. kube-proxy

c. Pod

d. All of the above

©Simplilearn. All rights reserved.

Knowledge Check 3	**Which of the following will run on Kubernetes nodes?**

a. Kubelet

b. kube-proxy

c. Pod

d. All of the above

The correct answer is **d. All of the above**

All the mentioned components run on Kubernetes nodes. They are interconnected and work together to host applications on pods.

©Simplilearn. All rights reserved.

Knowledge Check 4	**Which of the following is responsible for replication of controllers?**

a. ReplicationController

b. DaemonSet controller

c. Node controller

d. All of the above

©Simplilearn. All rights reserved.

Knowledge Check

4

Which of the following is responsible for replication of controllers?

a. ReplicationController

b. DaemonSet controller

c. Node controller

d. All of the above

The correct answer is **a. ReplicationController**

ReplicationController is responsible for replicating pods in case any node fails.

©Simplilearn. All rights reserved.

Lesson-End Project

Host Docker on a Kubernetes Cluster

Duration: 40 mins

Problem Statement:

Perform the following:

- Clone the GitHub repository to build the image.
- Confirm if any of the pods are running in the Kubernetes cluster.
- Run the Docker app at port 80.
- Host the Docker image built on a Kubernetes cluster.

Access: Click on the **Labs** tab on the left side panel of the LMS. Copy or note the username and password that is generated. Click on the **Launch Lab** button. On the page that appears, enter the username and password in their respective fields, and click **Login**.

©Simplilearn. All rights reserved.

Thank You

©Simplilearn. All rights reserved.