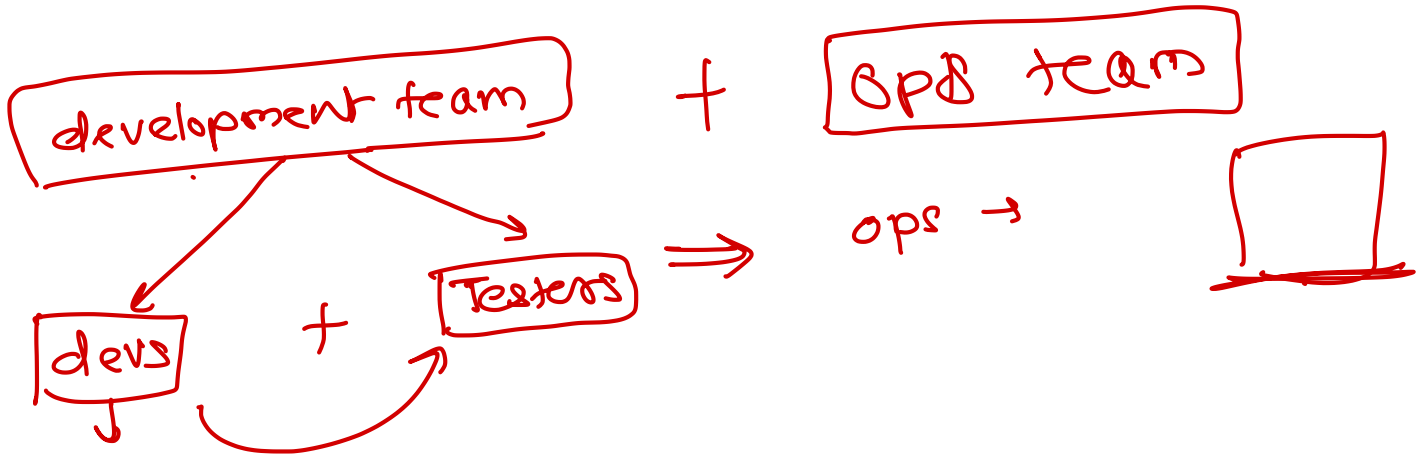
Jenkins

DevOps

Overview

- DevOps is a combination of two words development and operations
- Promotes collaboration between Development and Operations Team to deploy code to production faster in an automated & repeatable way
- DevOps helps to increases an organization's speed to deliver applications and services
- It allows organizations to serve their customers better and compete more strongly in the market
- Can be defined as an alignment of development and IT operations with better communication and collaboration



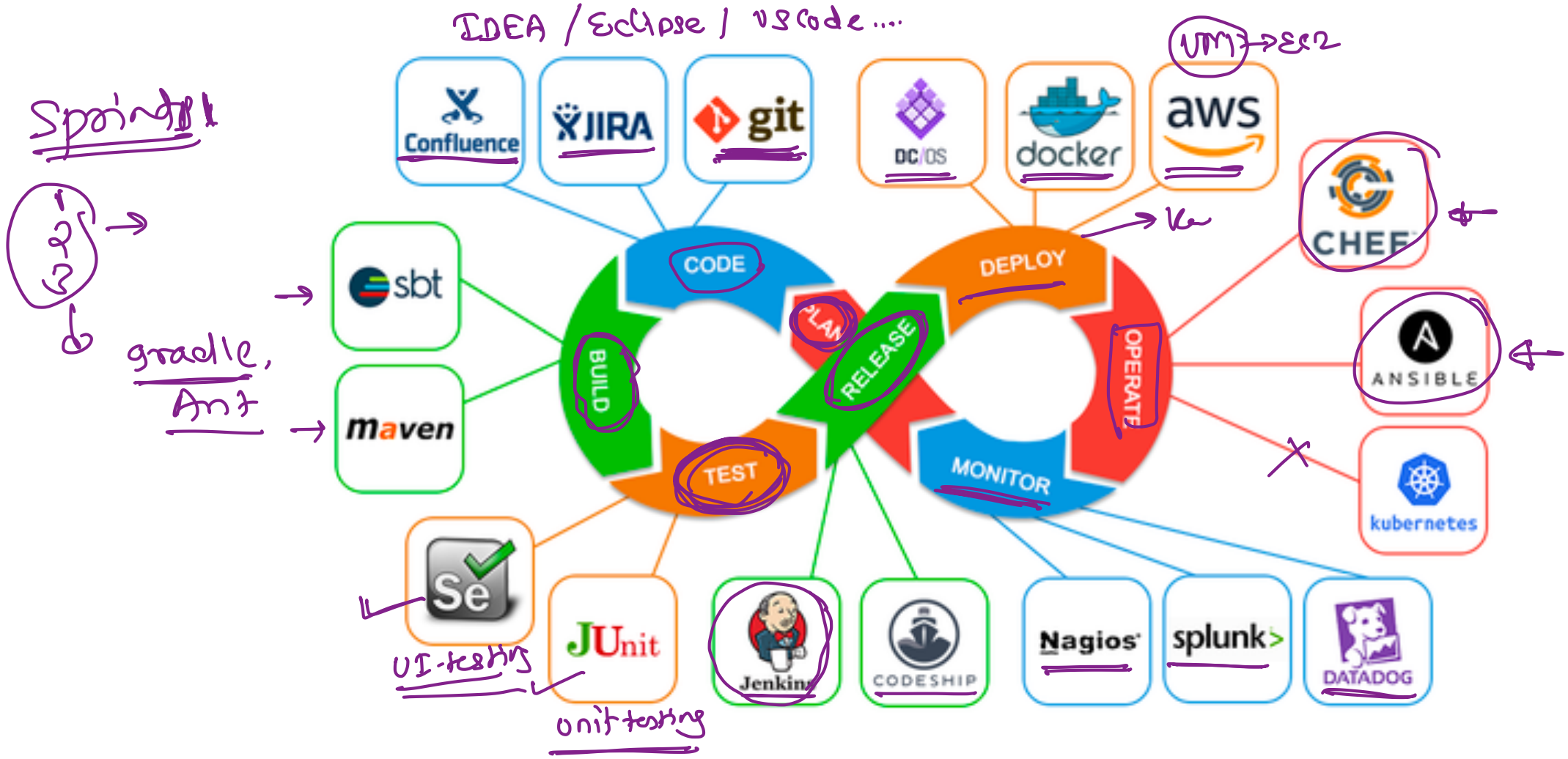
Why DevOps is Needed?

- Before DevOps, the development and operation team worked in complete isolation
- Testing and Deployment were isolated activities done after design-build. Hence they consumed more time than actual build cycles.
- Without using DevOps, team members are spending a large amount of their time in testing, deploying, and designing instead of building the project.
- Manual code deployment leads to human errors in production
- Coding & operation teams have their separate timelines and are not in synch causing further delays

What is DevOps ?

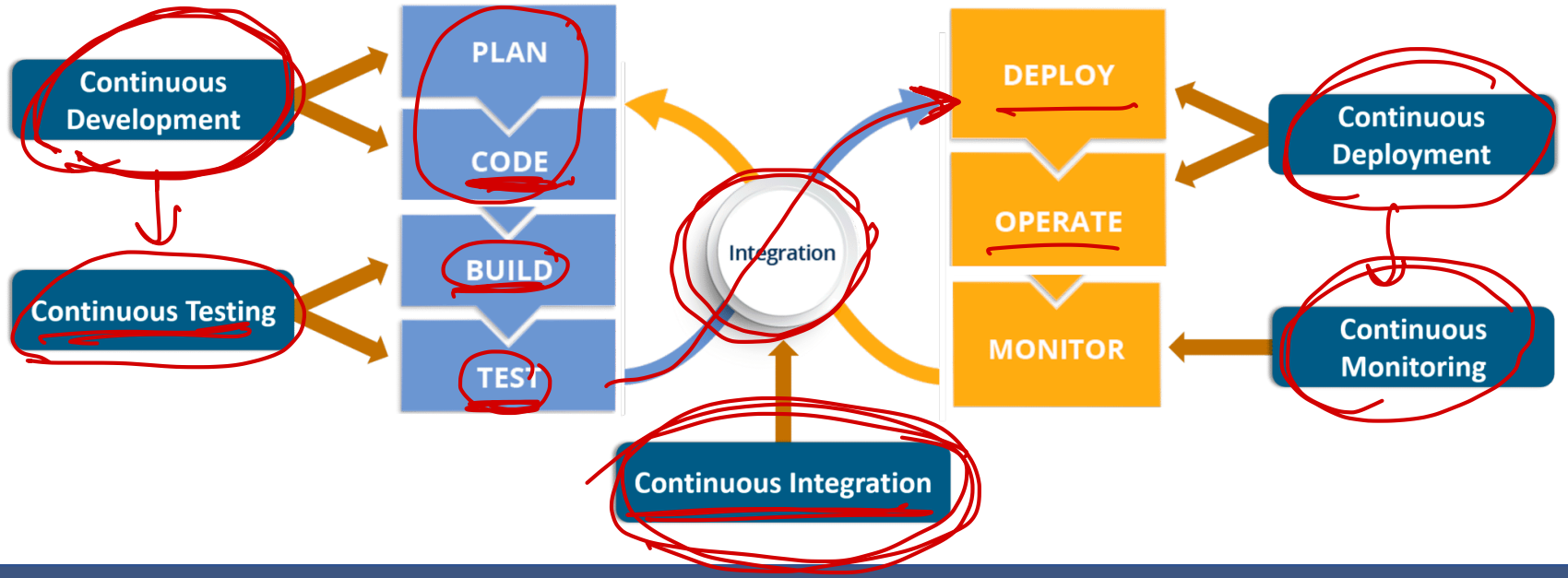
- DevOps is a never-ending process of continuous improvement
- It integrates Development and Operations teams
- It improves collaboration and productivity by
  - Automating infrastructure
  - Automating workflow
  - Continuously measuring application performance

DevOps Lifecycle



DevOps Terminologies

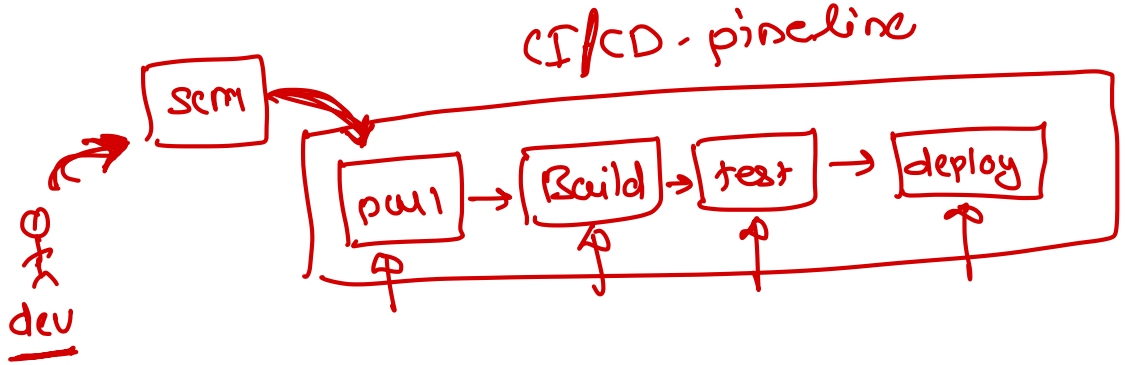
- Continuous Development
- Continuous Testing
- Continuous Integration
- Continuous Delivery
- Continuous Deployment
- Continuous Monitoring



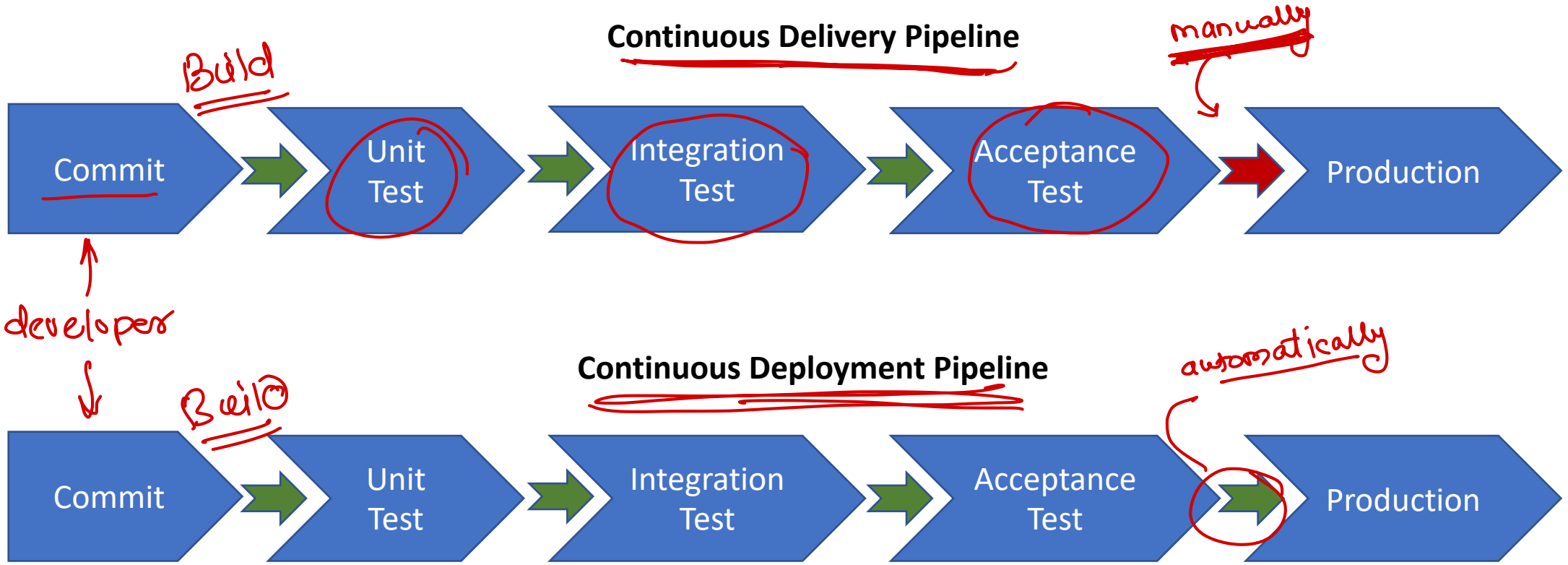
Continuous Integration

Overview

- It is the process of automating the building and testing of code, each time developer commits changes to the version control system
- CI is necessary to bring out issues encountered during the integration as early as possible
- CI requires developers to have frequent builds
- The common practice is that whenever a code commit occurs, a build should be triggered



CI/CD Pipeline



Importance

- Improves product quality
  - Improves the product quality by running the various unit test cases every time developer commits changes
- Increase productivity
  - Automating build of code saves a lot of time, thereby increasing productivity
  - Developer can utilize the time more to develop the code
- Reduces risk
  - Eliminates the potential human errors by automating test

Popular CI tools

Jenkins

TeamCity

Bamboo

GitLab CI

Travis CI


Jenkins

What is Jenkins ?

- Jenkins is a powerful application that allows continuous integration and continuous delivery of projects
- It is a free and open source application that can handle any kind of build or continuous integration

1. website
2. desktop app
3. mobile app
4. console app

Where is it came from ?

- It was first started as project Hudson at Sun Microsystems in 2004 and was first released in Feb 2005
- In 2010, Oracle acquired Sun Microsystems
- In 2011, Oracle created fork of Hudson as Jenkins, since when these two projects exist as two independent projects
- On April 20, 2016 version 2 was released with the *Pipeline* plugin enabled by default

Features

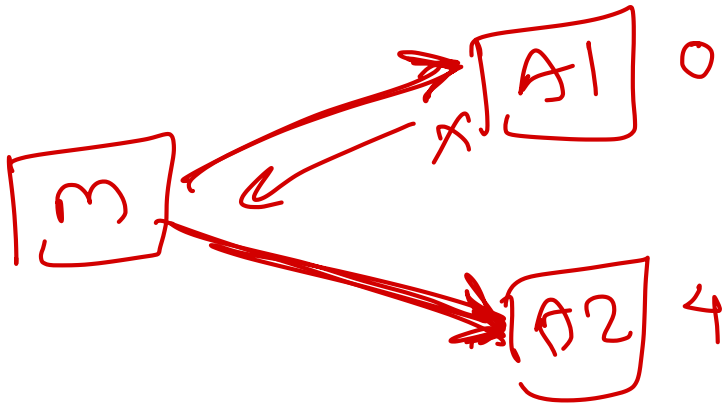
- Easy installation on different operating systems
- Supports pipelines as code that uses **domain-specific language** (**DSL**) to model application delivery pipelines as code
- Easily extensible with the use of third-party plugins
- Easy to configure the setup environment in the user interface
- Master slave architecture supports distributed builds to reduce the load on CI servers
- Build scheduling based on cron expressions
- Shell and Windows command execution that makes any command-line tool integration in the pipeline very easy
- Notification support related to build status

Terminologies

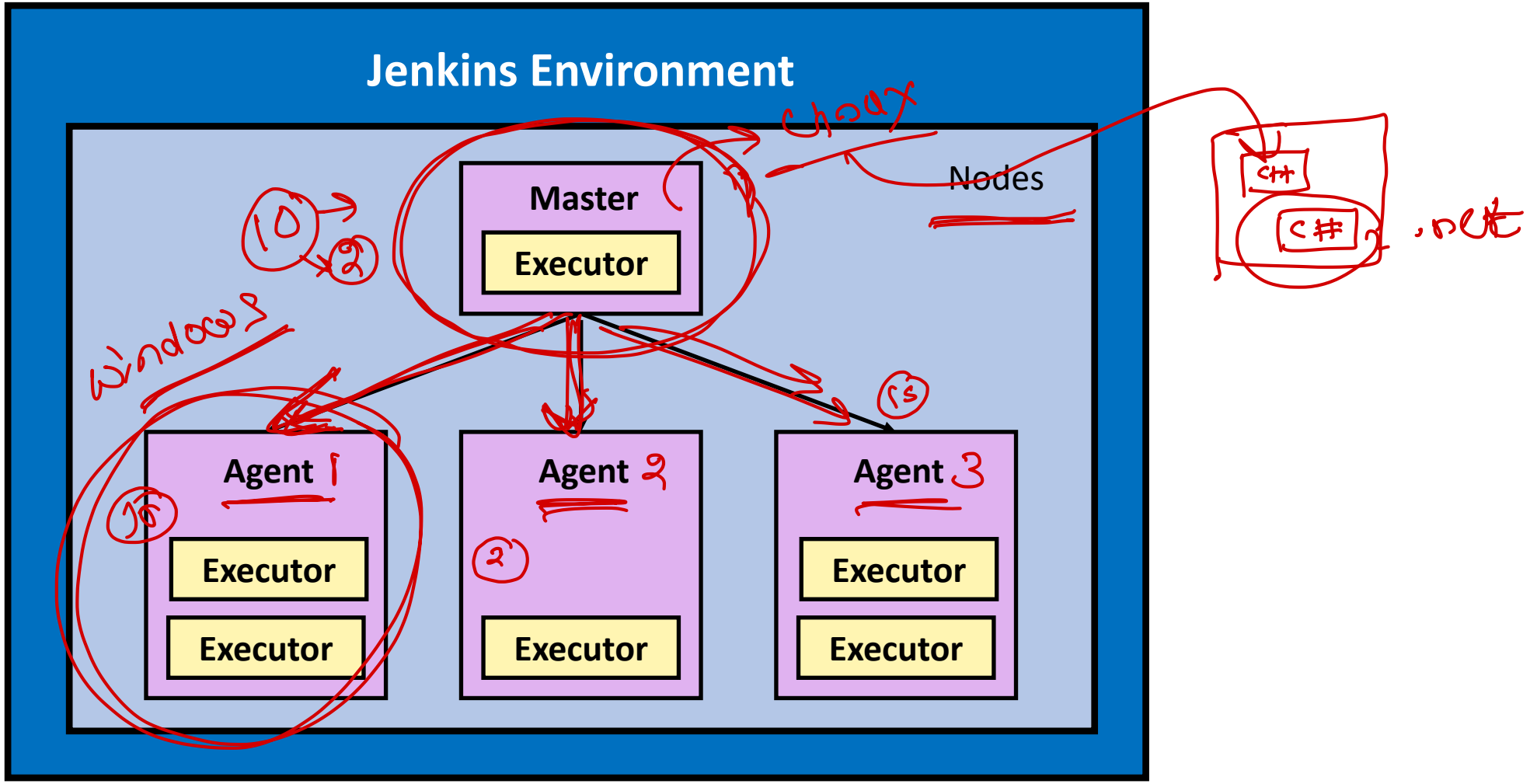
- **Node**
  - Node is the generic term that is used in Jenkins to mean any system that can run Jenkins jobs
  - This covers both masters and agents, and is sometimes used in place of those terms
  - Furthermore, a node might be a container, such as one for Docker
- **Master**
  - A Jenkins *master* is the primary controlling system for a Jenkins instance
  - It has complete access to all Jenkins configuration and options and the full list of jobs
  - It is the default location for executing jobs if another system is not specified
  - Master node must be present in Jenkins installation
- **Agent**
  - Is also known as Jenkins slave
  - This refers to any non-master system
  - The idea is that these systems are managed by the master system and allocated as needed, or as specified, to handle processing the individual jobs

Terminologies

- **Executor**
  - It is a slot in which to run a job on a node/agent
  - A node can have zero or more executors
  - The number of executors defines how many concurrent jobs can be run on that node
  - When the master funnels jobs to a particular node, there must be an available executor slot in order for the job to be processed immediately. Otherwise, it will wait until an executor becomes available.



Jenkins Environment



Job

- Also known as Project
- It represents the steps used to build the code
- To create a new job, use option “new item”
- Project in Jenkins has different types
  - Freestyle Project
  - Pipeline
  - Multi-configuration Project
  - Folder
  - GitHub Organization
  - Multibranch Pipeline