

Jenkins

Introduction and Basic Techniques



SoftUni Team
Technical Trainers



SoftUni



Software University

<https://softuni.bg>

You Have Questions?

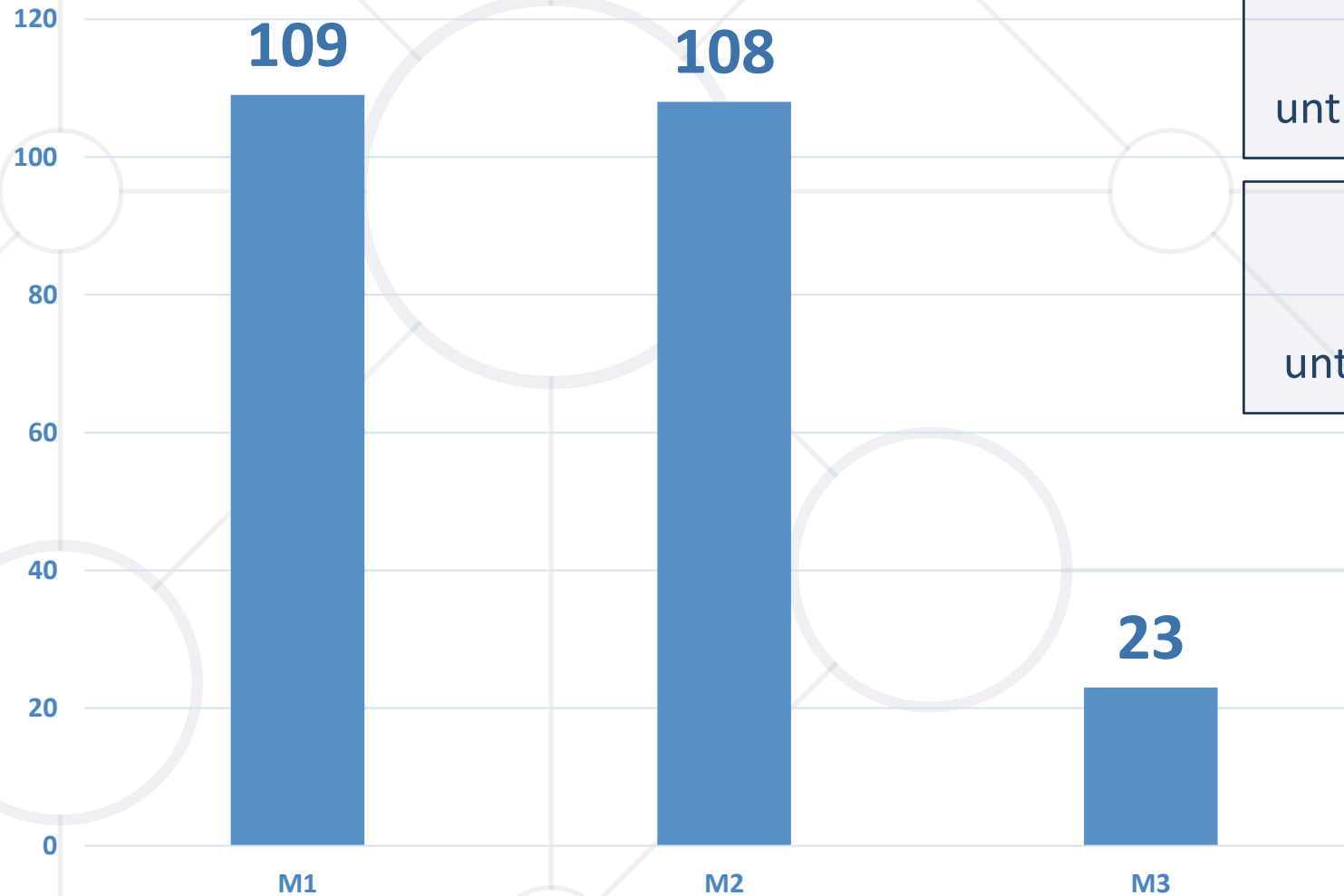
sli.do

#DevOps-23

facebook.com

/groups/DevOpsContainerizationCI/CDMonitoringJanuary2023

Homework Progress



Solutions for M3
can be submitted
until 23:59 on 22.02.2023

Solutions for M4
can be submitted
until 23:59 on 01.03.2023



Previous Module (M3)

Quick Overview

What We Covered

- Advanced techniques
 - Networking
 - Volumes
- Distributed Applications
 - Linking Methods
 - Docker Compose
- Docker Clusters
 - Components and Principles
 - Docker Swarm



This Module (M4)

Topics and Lab Infrastructure

1. Introduction to Jenkins
2. Working with Jenkins
 - Remote Tasks
 - Schedules
 - Plugins
3. Advanced Jenkins
 - Building with Docker
 - Pipelines



NETWORK: 192.168.99.0/24

.100

.101

VM1

Jenkins

VM2

Docker

VirtualBox

Vagrant

Windows / Linux / macOS



Available Solutions
For Continuous Integration/Deployment/Delivery

- **Speed**
 - Build frequently and faster
- **Automation**
 - Build and deploy in an automated fashion
- **Predictable results**
 - Test each step and deploy only if everything is okay
- **Faster time-to-market**
 - Possibility to deliver to production at any time

- **Buildbot**

- <http://buildbot.net/>
- Free

- **TeamCity**

- <https://www.jetbrains.com/teamcity/>
- Paid and Free. Cloud and On-Premise

- **Bamboo**

- <https://www.atlassian.com/software/bamboo>
- Only paid

- **CircleCI**

- <https://circleci.com/>
- Paid (offers basic free tier)

- **Travis CI**

- <https://travis-ci.com/>
- Free (and Paid)

- **Jenkins (Backed by CloudBees)**

- <https://jenkins.io/>
- Free (and Paid)

GitHub Actions

<https://github.com/features/actions>

Paid and free tier

- Continuous Delivery and Integration
 - CD is the ability to release at any time
 - CI is practice of integrating or merging code changes frequently
- Stages of CD and CI
 - Build => Deploy => Test => Release

Continuous Delivery

- Software **can be** released to production at any time
- Every change **can go** to production
- I **could be** deploying constantly

Continuous Deployment

- Software **is** released to production as part of an automated pipeline
- Every change **goes** to production
- I **am** deploying constantly





Introduction to Jenkins

Introduction. Components. Installation

What is Jenkins?

- An open source automation server
- A platform for the Software Development Life Cycle (SDLC)
- Typically used to implement CI/CD
- Easy to use and highly adaptable
- Extensible and customizable
- Works on most common operating systems
- Considered lightweight

- **Job**
 - Configured task in Jenkins. It is an old term
- **Project**
 - A task configured in Jenkins. It is the current term
- **Pipeline**
 - Special type of job created by a Pipeline plugin

- Requirements

- Works on Unix / Linux / Mac / Windows
- Requires Java 11 or 17

Current version

2.391 / 2.375.3

Weekly

LTS

- Installation

- Can be installed as a Native Service, Container, Java application
- Can be installed from source or through package system



Practice: Installation. Environment Setup
Live Demonstration in Class



Basic Activities

Remote Projects. Plugins. Schedules

- Local and Remote
- sudo allowance and settings
- Remote credentials
 - User and Password, Key file
- Firewall and SSH settings

- Default (minimal) set of plugins installed
- Over 1800 plugins available at <https://plugins.jenkins.io/>
- Grouped in categories by platform, purpose, etc.
- Dedicated Plugin Manager
- Automated or Manual installation

- With plugin or native
- Two options
 - Execute once at a specific point in time
 - Regular execution
- Execution interval is set with a cron like syntaxis

- Offered through a plugin
- Installed by default
- It works both with local git installation and GitHub
- Local git client is required
- Can receive notifications from git
- Can check periodically for changes (should be avoided)



Practice: See It in Action
Live Demonstration in Class



Advanced Jenkins #1

Docker Integration. Pipelines

- Integration can be done through plugin or directly
- Docker client is required
- Docker client should be present on all nodes where we plan to run Docker related tasks

- Old name is workflows
- Used for long running activity orchestration
- Can span on multiple nodes (slaves)
- It is a suite of plugins
- “Pipeline as code” is defined with a **Jenkinsfile**

- It is used to define continuous delivery pipeline
- Stored together with our code
- Two styles are supported – declarative and scripted
- It is written in Groovy

Declarative Jenkinsfile

It is required.

Specify which agent to run the pipeline or a stage. Can specify multiple nodes.

Mark the stages

Each stage describes a step in the SDLC

Steps are the actual work or actions to take place.
They are similar to “Build Step” in the Project Configuration view

Additional directives include: **post**, **environment**, **triggers**, and **parameters**

```
pipeline {  
  agents any  
  stages {  
    stage('Build') {  
      steps {  
        echo 'Step: Building...'  
      }  
    }  
    stage('Test') {  
      steps {  
        echo 'Step: Testing...'  
      }  
    }  
    stage('Deploy') {  
      steps {  
        echo 'Step: Deploying...'  
      }  
    }  
  }  
}
```



Practice: Slave Nodes and Docker

Live Demonstration in Class

- Jenkins is an open source automation server
- It could be installed in many ways
- It is extendable through plugins
- It is scalable – additional slave nodes could be added
- There is integration with source control systems
- Jenkins offers integration with Docker
- Jenkins could be managed with systems like Ansible, Chef, and etc.



Jenkins site

<https://jenkins.io/>

Jenkins installation guide

<https://jenkins.io/doc/book/installing/>

Jenkins Pipeline guide

<https://jenkins.io/doc/book/pipeline/>

Managing Jenkins

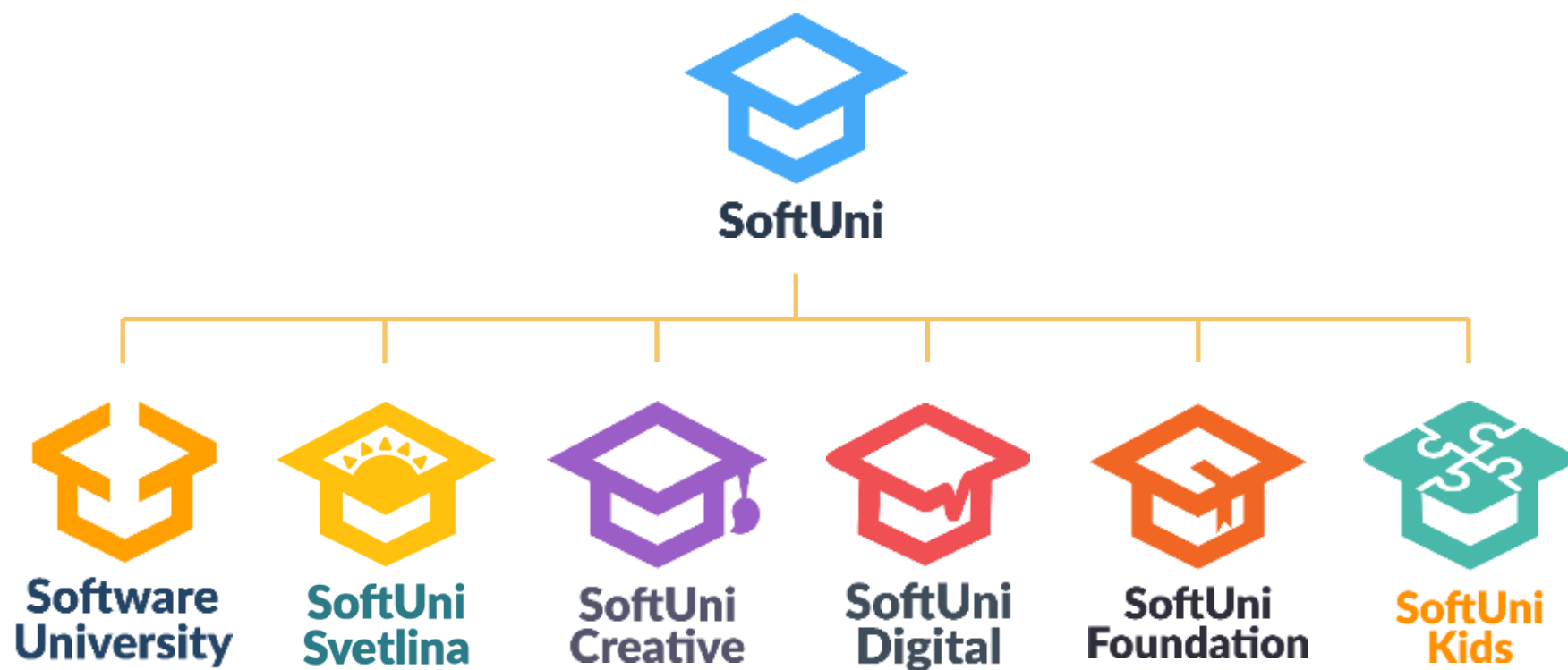
<https://jenkins.io/doc/book/managing/>

Jenkins plugins

<https://plugins.jenkins.io/>



Questions?



SoftUni Diamond Partners

SCHWARZ



Coca-Cola HBC
Bulgaria



Postbank

Решения за твоето утре



POKERSTARS



CAREERS



AMBITIONED

DXC
TECHNOLOGY



SOFTWARE
GROUP

Bosch.IO

INDEAVR
Serving the high achievers

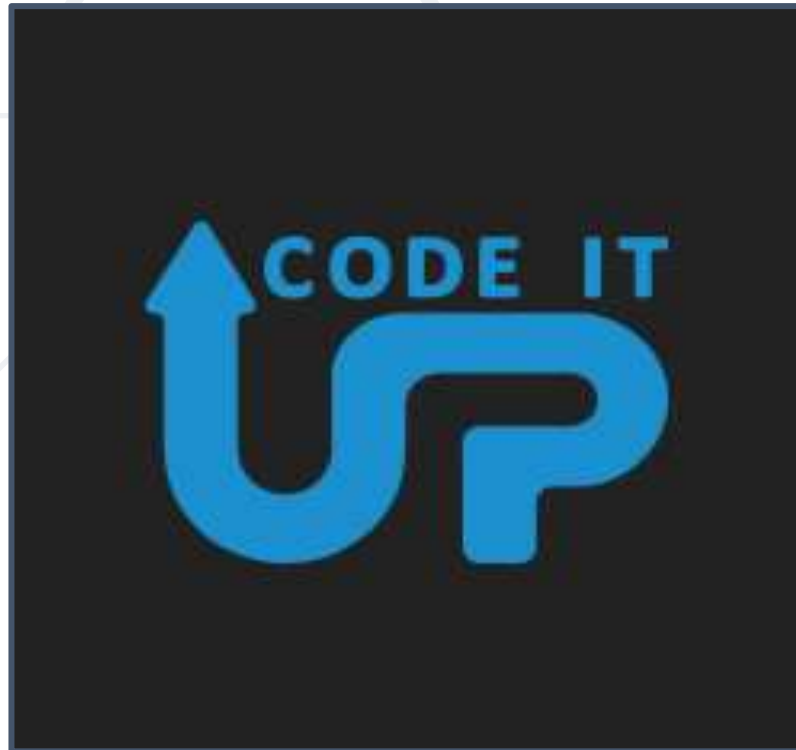
DRAFT
KINGS



SmartIT

createX

SUPER
HOSTING
.BG



- This course (slides, examples, demos, exercises, homework, documents, videos and other assets) is **copyrighted content**
- Unauthorized copy, reproduction or use is illegal
- © SoftUni – <https://softuni.org>
- © Software University – <https://softuni.bg>



- Software University – High-Quality Education, Profession and Job for Software Developers
 - softuni.bg, softuni.org
- Software University Foundation
 - softuni.foundation
- Software University @ Facebook
 - facebook.com/SoftwareUniversity
- Software University Forums
 - forum.softuni.bg

