IT 41 – DevOps

1. Define DevOps:

a) Development Operations

b) Development Options

c) Development and Operations Collaboration

d) Developer Operations

Answer: c) Development and Operations Collaboration

2. What is DevOps?

a) Software Development Methodology

b) Culture and Practice

c) Project Management Technique

d) None of the above

Answer: b) Culture and Practice

3. SDLC models, Lean, ITIL, Agile:

a) Sequential Development Life Cycle

b) Software Development Life Cycle

c) Systems Development Life Cycle

d) All of the above

Answer: b) Software Development Life Cycle

4. Why DevOps?

a) Faster Time to Market

b) Improved Deployment Frequency

c) Reduced Lead Time

d) All of the above

Answer: d) All of the above

5. History of DevOps:

a) Emerged in the 1990s

b) Emerged in the 2000s

c) Emerged in the 2010s

d) Emerged in the 2020s

Answer: b) Emerged in the 2000s

6. DevOps Stakeholders:

a) Developers only

b) Operations only

c) Both Developers and Operations

d) Project Managers only

Answer: c) Both Developers and Operations

7. DevOps Goals:

a) Increased Deployment Frequency

b) Faster Time to Detect Defects

c) Reduced Mean Time to Recovery

d) All of the above

Answer: d) All of the above

8. Important terminology:

a) Scrum

b) Kanban

c) Continuous Integration

d) All of the above

Answer: d) All of the above

9. DevOps perspective:

a) Siloed teams

b) Collaboration and Integration

c) Slow deployment

d) None of the above

Answer: b) Collaboration and Integration

10. DevOps and Agile:

a) They are the same

b) DevOps is an extension of Agile

c) Agile is an extension of DevOps

d) None of the above

Answer: b) DevOps is an extension of Agile

11. DevOps Tools:

a) Chef

b) Puppet

c) Jenkins

d) All of the above

Answer: d) All of the above

12. Configuration management:

a) Process of managing code

b) Process of managing infrastructure

c) Process of managing databases

d) Process of managing networks

Answer: b) Process of managing infrastructure

13. Continuous Integration and Deployment:

a) CI/CD

b) Continuous Development

c) Continuous Deployment

d) All of the above

Answer: a) CI/CD

14. Linux OS Introduction:

a) Linux is a proprietary operating system

b) Linux is an open-source operating system

c) Linux is developed by Microsoft

d) Linux is primarily used for gaming

Answer: b) Linux is an open-source operating system

15. Importance of Linux in DevOps:

a) Linux is not used in DevOps

b) Linux is essential for DevOps

c) Linux is only used for testing in DevOps

d) None of the above

Answer: b) Linux is essential for DevOps

16. Linux Basic Command Utilities:

a) ls, cd, mkdir

b) cat, grep, sed

c) cp, mv, rm

d) All of the above

Answer: d) All of the above

17. Linux Administration:

a) Managing user accounts

b) Installing software packages

c) Managing file permissions

d) All of the above

Answer: d) All of the above

18. Environment Variables:

a) Variables defined for the current user session

b) Variables defined for the entire system

c) Variables used only by applications

d) None of the above

Answer: a) Variables defined for the current user session

19. Networking:

a) Involves connecting computers and devices

b) Involves managing software configurations

c) Involves optimizing code performance

d) None of the above

Answer: a) Involves connecting computers and devices

20. Linux Server Installation:

a) GUI-based installation only

b) Command-line installation only

c) Both GUI-based and command-line installation

d) None of the above

Answer: c) Both GUI-based and command-line installation

21. RPM and YUM Installation:

a) RPM is a package manager for Ubuntu

b) YUM is a package manager for Debian

c) RPM and YUM are package managers for CentOS and Fedora

d) None of the above

Answer: c) RPM and YUM are package managers for CentOS and Fedora

22. Introduction to GIT:

a) GIT is a version control system

b) GIT stands for Global Information Tracker

c) GIT is used primarily for project management

d) All of the above

Answer: a) GIT is a version control system

23. What is Git:

a) Distributed version control system

b) Centralized version control system

c) Personal version control system

d) None of the above

Answer: a) Distributed version control system

24. About Version Control System and Types:

a) VCS manages changes to documents

b) There are two types of VCS: Centralized and Distributed

c) GIT is a centralized VCS

d) None of the above

Answer: b) There are two types of VCS: Centralized and Distributed

25. Difference between CVCS and DVCS:

a) CVCS has a single central repository, while DVCS has multiple repositories

b) CVCS is faster than DVCS

c) DVCS requires an internet connection, while CVCS does not

d) None of the above

Answer: a) CVCS has a single central repository, while DVCS has multiple repositories

26. A short history of GIT:

a) Created by Linus Torvalds in 2005

b) Created by Bill Gates in 1990

c) Created by Steve Jobs in 2007

d) None of the above

Answer: a) Created by Linus Torvalds in 2005

27. GIT Basics:

a) Clone, add, commit, push, pull

b) Create, update, delete, merge, revert

c) Open, close, save, export

d) None of the above

Answer: a) Clone, add, commit, push, pull

28. GIT Command Line:

a) GUI-based interface for GIT

b) Command-line interface for GIT

c) Web-based interface for GIT

d) None of the above

Answer: b) Command-line interface for GIT

29. Installing Git:

a) Git is pre-installed on all operating systems

b) Git can be installed using package managers

c) Git can only be installed manually from source

d) None of the above

Answer: b) Git can be installed using package managers

30. Installing on Linux:

a) sudo apt-get install git

b) sudo yum install git

c) sudo pacman -S git

d) All of the above

Answer: d) All of the above

31. Installing on Windows:

a) Download and run the installer from the Git website

b) Install using Chocolatey package manager

c) Use Windows Subsystem for Linux (WSL) to install Git

d) None of the above

Answer: a) Download and run the installer from the Git website

32. Initial setup:

a) Configure username and email

b) Configure password and token

c) Configure SSH key

d) All of the above

Answer: a) Configure username and email

33. Git Essentials:

a) Branching, merging, rebasing

b) Cloning, forking, tagging

c) Pushing, pulling, fetching

d) All of the above

Answer: d) All of the above

34. Creating repository:

a) git create

b) git init

c) git new

d) None of the above

Answer: b) git init

35. Cloning, check-in and committing:

a) git clone, git check-in, git commit

b) git pull, git push, git merge

c) git clone, git add, git commit

d) None of the above

Answer: c) git clone, git add, git commit

36. Fetch pull and remote:

a) git fetch, git pull, git remote

b) git fetch, git push, git pull

c) git clone, git fetch, git pull

d) None of the above

Answer: a) git fetch, git pull, git remote

37. Branching:

a) Creating new branches for development

b) Merging branches back into the main branch

c) Rebasing branches onto the main branch

d) All of the above

Answer: d) All of the above

38. Creating the Branches, switching the branches, merging:

a) git branch, git switch, git merge

b) git create, git switch, git join

c) git branch, git checkout, git merge

d) None of the above

Answer: c) git branch, git checkout, git merge

39. The branches:

a) Local and Remote

b) Main and Feature

c) Master and Dev

d) All of the above

Answer: a) Local and Remote

40. Overview of Chef; Common Chef Terminology:

a) Chef is a version control system

b) Chef is a configuration management tool

c) Chef uses YAML syntax

d) None of the above

Answer: b) Chef is a configuration management tool

41. Workstation Setup: How to configure knife Execute:

a) knife configure

b) knife setup

c) knife configure initial

d) None of the above

Answer: a) knife configure

42. Organization Setup: Create organization; Add yourself and node to organization:

a) chef create organization

b) chef org create

c) knife org create

d) None of the above

Answer: c) knife org create

43. Test Node Setup: Create a server and add to organization, check node details using knife:

a) knife node create

b) knife node setup

c) knife node test

d) None of the above

Answer: a) knife node create

44. Node Objects and Search: How to Add Run list to Node Check node Details:

a) knife node run_list

b) knife node list

c) knife node edit

d) None of the above

Answer: c) knife node edit

45. Environments: How to create Environments, Add servers to environments:

a) chef create environment

b) knife environment create

c) chef env create

d) None of the above

Answer: b) knife environment create

46. Roles: Create roles, Add Roles to organization:

a) knife role create

b) chef role create

c) knife create role

d) None of the above

Answer: a) knife role create

47. Attributes: Understanding of Attributes, Creating Custom Attributes, Defining in Cookbooks:

a) Attributes are metadata about a node

b) Attributes can be defined in recipes

c) Attributes can be overridden at different levels

d) All of the above

Answer: d) All of the above

48. Data bags: Understanding the data bags, Creating and managing the Data bags, Creating the data bags using CLI and Chef Console, Sample Data bags for Creating Users:

a) Data bags are used to store sensitive information

b) Data bags can be encrypted

c) Data bags are accessed using knife commands

d) All of the above

Answer: d) All of the above

49. Maven Installation:

a) Download and unzip Maven

b) Use package manager to install Maven

c) Maven is pre-installed on all operating systems

d) None of the above

Answer: a) Download and unzip Maven

50. Maven Build requirements:

a) Maven requires Java Development Kit (JDK)

b) Maven requires Python

c) Maven requires Ruby

d) None of the above

Answer: a) Maven requires Java Development Kit (JDK)

51. Maven POM Builds (pom.xml):

a) POM stands for Project Object Model

b) pom.xml is the configuration file for Maven

c) POM defines project dependencies, plugins, and build settings

d) All of the above

Answer: d) All of the above

52. Maven Build Life Cycle:

a) Clean, Compile, Package, Install, Deploy

b) Validate, Compile, Test, Package, Verify, Install, Deploy

c) Initialize, Build, Deploy

d) None of the above

Answer: b) Validate, Compile, Test, Package, Verify, Install, Deploy

53. Maven Local Repository (.m2):

a) Default location ~/.m2/repository

b) Used to store project dependencies

c) Can be configured in settings.xml

d) All of the above

Answer: d) All of the above

54. Maven Global Repository:

a) Default location ~/.m2/repository

b) Used to store project dependencies

c) Can be configured in settings.xml

d) None of the above

Answer: b) Used to store project dependencies

55. Group ID, Artifact ID, Snapshot:

a) Group ID identifies the project group

b) Artifact ID identifies the project

c) Snapshot denotes a development version

d) All of the above

Answer: d) All of the above

56. Maven Dependencies:

a) Dependencies are declared in the POM file

b) Dependencies are resolved transitively

c) Maven downloads dependencies from central repository by default

d) All of the above

Answer: d) All of the above

57. Maven Plugins:

a) Plugins extend Maven's functionality

b) Plugins are configured in the POM file

c) Plugins are executed during the build lifecycle

d) All of the above

Answer: d) All of the above

58. Introduction: What is Docker, Use case of Docker, Platforms for Docker, Docker vs. Virtualization:

a) Docker is a containerization platform

b) Docker is used for packaging applications

c) Docker containers are lightweight and portable

d) All of the above

Answer: d) All of the above

59. Architecture: Docker Architecture., Understanding the Docker components:

a) Docker Engine, Docker Hub, Docker Swarm

b) Docker Client, Docker Daemon, Docker Registry

c) Docker Containers, Docker Images, Dockerfile

d) All of the above

Answer: b) Docker Client, Docker Daemon, Docker Registry

60. Installation: Installing Docker on Linux. Understanding Installation of Docker on windows. Some Docker commands. Provisioning:

a) Docker can be installed using package managers

b) Docker can be installed from source

c) Docker commands include docker run, docker build, docker push

d) All of the above

Answer: d) All of the above

61. Docker Hub.: Downloading Docker images. Uploading the images in Docker Registry and AWS ECS, Understanding the containers, Running commands in container. Running multiple containers:

a) Docker Hub is a public registry for Docker images

b) Docker Registry is a private registry for Docker images

c) Docker images can be pushed to AWS ECS

d) All of the above

Answer: d) All of the above

62. Custom images: Creating a custom image. Running a container from the custom image. Publishing the custom image:

a) Custom images are created using Dockerfiles

b) Custom images can be tagged with a version

c) Custom images can be pushed to Docker Hub

d) All of the above

Answer: d) All of the above

63. Docker Networking: Accessing containers, linking containers, Exposing container ports, Container Routing:

a) Docker containers can communicate with each other using networking

b) Containers can be linked to share data and resources

c) Container ports can be exposed to the host system

d) All of the above

Answer: d) All of the above