General Questions on Azure DevOps Services

1. **What is Azure DevOps?**
   A. A cloud storage solution
   B. A version control platform
   C. A set of development tools and services
   D. A deployment pipeline tool
   Answer: C
   Explanation: Azure DevOps is a suite of tools and services that enable software development, continuous integration/continuous deployment (CI/CD), project management, and collaboration.
2. **Which of the following services is NOT part of Azure DevOps?**
   A. Azure Pipelines
   B. Azure Repos
   C. Azure Tables
   D. Azure Test Plans
   Answer: C
   Explanation: Azure Tables is part of Azure Storage, not Azure DevOps. Azure DevOps includes Pipelines, Repos, Boards, Test Plans, and Artifacts.
3. **What is the purpose of Azure Pipelines?**
   A. To manage project tasks
   B. To provide CI/CD workflows
   C. To host private Git repositories
   D. To monitor application health
   Answer: B
   Explanation: Azure Pipelines facilitates CI/CD, enabling automated builds, tests, and deployments.
4. **Which command-line interface can you use with Azure DevOps?**
   A. Azure CLI
   B. Git CLI
   C. az pipelines CLI
   D. All of the above
   Answer: D
   Explanation: Azure DevOps integrates with Azure CLI, Git CLI, and specific `az pipelines` commands for managing pipelines and deployments.
5. **Which Azure DevOps service is used for backlog tracking and sprint planning?**
   A. Azure Boards
   B. Azure Repos
   C. Azure Artifacts
   D. Azure Pipelines
   Answer: A
   Explanation: Azure Boards provides features for agile project management like backlog tracking, sprint planning, and Kanban boards.

Scenario-Based Questions

6. **You want to enable team members to collaborate on code changes while reviewing code quality. Which service should you use?**
   A. Azure Boards
   B. Azure Pipelines
   C. Azure Repos
   D. Azure Artifacts
   Answer: C
   Explanation: Azure Repos hosts Git repositories and facilitates collaboration through pull requests, branch policies, and code reviews.
7. **Your application must automatically deploy to staging after passing all unit tests. Which Azure DevOps feature should you implement?**
   A. Deployment Triggers
   B. Release Pipelines
   C. Test Plans
   D. Artifacts Feed
   Answer: B
   Explanation: Release Pipelines support deploying applications to different environments, such as staging or production, after successful tests.
8. **A build fails due to missing dependencies. How can Azure DevOps ensure dependencies are available for future builds?**
   A. Use Azure Boards
   B. Host dependencies in Azure Artifacts
   C. Store them in a Git repository
   D. Add them to the pipeline YAML
   Answer: B
   Explanation: Azure Artifacts provides a universal package repository to store and manage dependencies like NuGet, npm, or Maven packages.
9. **A team wants to ensure only tested code is merged into the main branch. Which approach should they use?**
   A. Commit directly to the main branch
   B. Create pull requests with branch policies
   C. Set up deployment gates
   D. Run manual tests on the branch
   Answer: B
   Explanation: Branch policies enforce quality by requiring pull requests, code reviews, and automated tests before merging.
10. **How can you monitor the progress of multiple builds across different pipelines in real-time?**
    A. Azure Boards
    B. Azure DevOps Dashboard
    C. Azure CLI Logs
    D. Azure Portal Overview
    Answer: B
    Explanation: The Azure DevOps Dashboard can display widgets and charts to monitor pipeline progress, build results, and other metrics in real-time.

Pipeline & CI/CD Questions

11. **What file format is commonly used to define Azure Pipelines?**
    A. JSON
    B. YAML
    C. XML
    D. CSV
    Answer: B
    Explanation: Azure Pipelines uses YAML (Yet Another Markup Language) files to define pipeline workflows, including build, test, and deployment steps.
12. **Which trigger automatically starts a pipeline when code is pushed?**
    A. Scheduled trigger
    B. Pull request trigger
    C. CI trigger
    D. Manual trigger
    Answer: C
    Explanation: A Continuous Integration (CI) trigger automatically initiates a build when changes are pushed to the repository.
13. **What is a primary benefit of using a multi-stage pipeline?**
    A. Faster builds
    B. Reduced costs
    C. Clear segregation of environments
    D. Improved logging
    Answer: C
    Explanation: Multi-stage pipelines allow developers to define and separate build, test, and deployment environments clearly.
14. **How can you pass variables securely in a pipeline?**
    A. Hard-code them in YAML
    B. Use a variable group with secrets enabled
    C. Add them to the Git repository
    D. Store them in Azure Boards
    Answer: B
    Explanation: Variable groups with secrets enabled in Azure Pipelines securely store sensitive information like API keys or passwords.
15. **What does the `pool` keyword in a pipeline YAML file specify?**
    A. Test configurations
    B. Repository location
    C. Agent where the job runs
    D. Deployment target
    Answer: C
    Explanation: The `pool` keyword specifies the agent pool to use for running the pipeline's jobs.

Scenario-Based Advanced Questions

16. **Your deployment requires approval before moving to production. What feature would you use?**
    A. Manual Intervention Task
    B. Release Gate Approvals
    C. Test Plans
    D. Azure Monitor Alerts
    Answer: B
    Explanation: Release Gate Approvals ensure a manual or automated approval step before proceeding to production.
17. **Your build pipeline needs to run specific tests only when files in the `tests/` directory are modified. What should you configure?**
    A. Build triggers
    B. Path filters
    C. Deployment conditions
    D. Task filters
    Answer: B
    Explanation: Path filters define specific directories or files that trigger a pipeline run when modified.
18. **A new developer needs read-only access to a pipeline. Which role should you assign?**
    A. Administrator
    B. Contributor
    C. Reader
    D. Owner
    Answer: C
    Explanation: The Reader role provides view-only access, ensuring no changes can be made to the pipeline.
19. **You want to visualize work item progress and deployment success rates on a single page. What tool would you use?**
    A. Analytics View
    B. Azure DevOps Dashboard
    C. Test Plan Metrics
    D. Pipeline Logs
    Answer: B
    Explanation: Dashboards in Azure DevOps aggregate work item, pipeline, and other data in customizable widgets for easy visualization.
20. **How can you enforce a build validation policy for a branch in Azure DevOps?**
    A. Add a pre-merge validation pipeline
    B. Configure a branch policy with a build pipeline
    C. Add manual reviewers for pull requests
    D. Use deployment gates
    Answer: B
    Explanation: Branch policies with build validation ensure that code merges into the branch only after passing a specified build pipeline.

---

21. **What is the primary purpose of Azure Artifacts?**
    A. Store source code
    B. Provide CI/CD templates
    C. Host packages like NuGet, npm, or Maven
    D. Test application performance
    Answer: C
    Explanation: Azure Artifacts is a package management solution that hosts and shares packages for developers to integrate into their pipelines.
22. **Which tool in Azure DevOps is used for managing manual and automated test cases?**
    A. Azure Boards
    B. Azure Pipelines
    C. Azure Test Plans
    D. Azure Artifacts
    Answer: C
    Explanation: Azure Test Plans allows teams to manage test cases, execute manual and exploratory tests, and integrate automated tests with pipelines.
23. **What does a "release" in Azure Pipelines typically represent?**
    A. A completed feature
    B. A deployment to an environment
    C. A Git commit
    D. A rollback operation
    Answer: B
    Explanation: A release in Azure Pipelines represents the deployment of an application to an environment such as staging or production.
24. **Which Azure DevOps service provides Kanban and Scrum support?**
    A. Azure Boards
    B. Azure Pipelines
    C. Azure Artifacts
    D. Azure Monitor
    Answer: A
    Explanation: Azure Boards supports agile project management methodologies, including Kanban and Scrum.
25. **What is the significance of the "agent" in Azure Pipelines?**
    A. It schedules tasks
    B. It executes pipeline jobs
    C. It monitors deployment status
    D. It stores artifacts
    Answer: B
    Explanation: An agent is a virtual machine or container that executes pipeline jobs by running the build and deployment tasks.

Scenario-Based Questions (Continued)

26. **You need to implement a workflow that requires database migrations to succeed before deploying an application. What feature should you use?**
    A. Deployment conditions
    B. Pre-deployment approvals
    C. Post-deployment approvals
    D. Multi-stage pipelines
    Answer: D
    Explanation: Multi-stage pipelines allow you to define separate stages for tasks like database migrations and application deployment with conditional execution.
27. **Your team wants to track how long features take to move from development to production. Which Azure DevOps feature can help?**
    A. Dashboards
    B. Analytics views
    C. Work Item Age Chart
    D. Sprint Velocity Chart
    Answer: B
    Explanation: Analytics views can provide insights into lead time, cycle time, and other metrics for work items.
28. **A build task intermittently fails due to network issues. How can you make the pipeline more resilient?**
    A. Increase task timeout
    B. Use retry policies
    C. Split the pipeline into smaller jobs
    D. Manually retry failed builds
    Answer: B
    Explanation: Retry policies in Azure Pipelines can automatically reattempt tasks that fail due to transient issues like network instability.
29. **A pipeline must only run after a manual trigger. Which configuration should you apply?**
    A. CI trigger
    B. Scheduled trigger
    C. Pull request trigger
    D. Disable triggers and run manually
    Answer: D
    Explanation: Disabling all automatic triggers ensures the pipeline only runs when manually initiated.
30. **How can you restrict who can approve production deployments in Azure DevOps?**
    A. Configure role-based access control (RBAC)
    B. Use branch policies
    C. Set up gated builds
    D. Add specific approvers to release gates
    Answer: D
    Explanation: Release gates allow you to define specific approvers for production deployments to ensure proper authorization.

Advanced Topics in Azure DevOps

31. **What is the purpose of conditional expressions in a pipeline?**
    A. To handle parallel tasks
    B. To customize task execution based on conditions
    C. To trigger builds automatically
    D. To control artifact retention policies
    Answer: B
    Explanation: Conditional expressions allow tasks to run only if specific criteria are met, such as environment variables or build results.
32. **Which authentication method is recommended for securing pipeline secrets?**
    A. Hardcoding secrets in YAML
    B. Using Azure Key Vault
    C. Storing secrets in Git
    D. Adding secrets to the dashboard
    Answer: B
    Explanation: Azure Key Vault securely stores and manages sensitive information like API keys and secrets, which pipelines can access at runtime.
33. **How can you visualize build and release metrics over time?**
    A. Azure DevOps Dashboards
    B. Test Plans
    C. Azure Monitor
    D. Git Insights
    Answer: A
    Explanation: Dashboards provide widgets and charts to visualize pipeline performance, build results, and other metrics.
34. **You want to reduce deployment downtime for a production environment. Which deployment strategy should you use?**
    A. Blue-green deployment
    B. Manual deployment
    C. Incremental deployment
    D. Test-first deployment
    Answer: A
    Explanation: Blue-green deployment involves deploying a new version of the application to a separate environment and switching traffic to it, minimizing downtime.
35. **Which of the following allows you to integrate external issue trackers like Jira with Azure DevOps?**
    A. Service Hooks
    B. Deployment Gates
    C. Analytics Views
    D. YAML Templates
    Answer: A
    Explanation: Service Hooks allow Azure DevOps to integrate with external tools like Slack, Jira, and Jenkins for event-based notifications.

Scenario-Based Questions on Dashboards

36. **A team needs to display a chart of resolved bugs vs. created bugs on a sprint dashboard. Which widget should they use?**
    A. Sprint Progress
    B. Work Item Chart
    C. Velocity Chart
    D. Code Coverage
    Answer: B
    Explanation: The Work Item Chart widget visualizes metrics like created and resolved bugs, making it suitable for this scenario.
37. **Your dashboard must include pipeline run statuses and the time taken for each build. What should you add?**
    A. Work Item Chart widget
    B. Pipeline Widget
    C. Release Overview widget
    D. Code Coverage widget
    Answer: B
    Explanation: The Pipeline Widget displays build statuses, durations, and results directly on the dashboard.
38. **Which widget is most useful for tracking sprint velocity?**
    A. Query Results
    B. Velocity Chart
    C. Build History
    D. Release Summary
    Answer: B
    Explanation: The Velocity Chart widget helps teams visualize completed story points or work items across sprints.
39. **You want to monitor flaky tests in a pipeline over time. Which report should you consult?**
    A. Test Trends
    B. Code Coverage
    C. Test Plan Overview
    D. Deployment Logs
    Answer: A
    Explanation: Test Trends provide insights into test reliability and identify flaky tests by analyzing results over multiple runs.
40. **How can you track whether a pipeline has successfully deployed code in recent runs?**
    A. View Deployment Gates
    B. Monitor the Pipeline History widget
    C. Use Test Plan dashboards
    D. Analyze release logs
    Answer: B
    Explanation: The Pipeline History widget shows the success or failure of recent pipeline runs.

Advanced Technical Questions on Azure DevOps

41. **What happens if you define multiple pipelines in separate YAML files in the same repository?**
    A. Only the first pipeline will execute.
    B. All pipelines will trigger simultaneously on a commit.
    C. Each pipeline runs based on its triggers and path filters.
    D. Pipelines will conflict and fail execution.
    Answer: C
    Explanation: Azure DevOps allows multiple YAML pipelines in a single repository, with each executing independently based on its defined triggers, including branch and path filters.
42. **How can you ensure that a pipeline job executes on a self-hosted agent instead of a Microsoft-hosted agent?**
    A. Use the `pool` keyword and specify the agent pool name.
    B. Change the pipeline YAML schema to self-hosted mode.
    C. Configure a custom pipeline trigger.
    D. Define the agent type in Azure Boards.
    Answer: A
    Explanation: The `pool` keyword in the YAML pipeline allows you to specify whether the job should run on a Microsoft-hosted agent or a self-hosted agent by naming the desired agent pool.
43. **You want to run a task only if the previous task fails. How can you configure this in Azure Pipelines?**
    A. Add a conditional `dependsOn` statement.
    B. Use the `condition: failed()` directive.
    C. Set the `always()` condition.
    D. Manually trigger the task upon failure.
    Answer: B
    Explanation: The `condition: failed()` directive ensures the task runs only if the preceding task fails, allowing flexible error handling in the pipeline.
44. **How can you share variables across multiple pipelines within the same Azure DevOps project?**
    A. Define variables in each YAML file.
    B. Use a variable group in the Library.
    C. Add a variable template to each pipeline.
    D. Pass variables through the command line.
    Answer: B
    Explanation: Variable groups in Azure DevOps Library enable sharing of variables, including secrets, across multiple pipelines.
45. **What happens when a pipeline uses a `YAML template` but the template file is updated during a run?**
    A. The pipeline restarts automatically with the updated template.
    B. The pipeline continues using the previously cached version of the template.
    C. The pipeline fails due to version mismatch.
    D. The updated template applies immediately for subsequent jobs.
    Answer: D

Explanation: YAML templates in Azure Pipelines are applied dynamically, but the updated template takes effect only in future pipeline runs after the commit.

Scenario-Based Pipeline Challenges

46. **You need to dynamically deploy a microservice to a Kubernetes cluster only when its source code changes. What configuration ensures this?**
    A. Add path filters for the microservice directory in the pipeline trigger.
    B. Use environment variables to detect changes.
    C. Implement deployment gates in Azure Release Pipelines.
    D. Use manual intervention steps before deployment.
    Answer: A
    Explanation: Path filters in pipeline triggers ensure that the pipeline only runs when specific files or directories, such as the microservice's code, are modified.
47. **A pipeline fails intermittently on a task due to API rate limits. What is the best approach to mitigate this?**
    A. Retry the entire pipeline manually.
    B. Add a retry policy with a delay to the failing task.
    C. Increase the agent's timeout setting.
    D. Split the task into smaller subtasks.
    Answer: B
    Explanation: A retry policy with delays can handle transient failures, such as API rate limits, without manual intervention.
48. **You need to ensure the database schema is updated before deploying an application to production. What pipeline feature can you use?**
    A. Add a Pre-deployment Condition.
    B. Use Deployment Stages with approvals.
    C. Create a separate build pipeline for database updates.
    D. Use a deployment gate with custom conditions.
    Answer: B
    Explanation: Deployment Stages in multi-stage pipelines allow you to define steps like schema updates as part of the deployment workflow, ensuring they execute sequentially.
49. **How do you pass secure values, such as API keys, into a pipeline while keeping them hidden?**
    A. Store them as environment variables in the YAML file.
    B. Add them as secrets in a variable group or pipeline variables.
    C. Include them directly in the Git repository.
    D. Encode them as base64 in the YAML file.
    Answer: B
    Explanation: Secrets stored in variable groups or pipeline variables remain encrypted and hidden in logs, ensuring security during pipeline execution.
50. **You need to deploy an application to multiple regions simultaneously while sharing build artifacts. What is the recommended approach?**
    A. Create separate pipelines for each region.
    B. Use deployment jobs with parallel execution.

C. Configure manual triggers for each region.
D. Build separate artifacts for each deployment.
Answer: B
Explanation: Deployment jobs with parallel execution allow you to deploy the same build artifact to multiple regions simultaneously.

Advanced Questions on Azure DevOps Dashboard

51. **How can you customize the Azure DevOps Dashboard for different team members?**
    A. Create separate dashboards for each team member.
    B. Use widget filters to personalize data for specific users.
    C. Restrict dashboard permissions to administrators only.
    D. Share a single read-only dashboard for all users.
    Answer: B
    Explanation: Widget filters allow users to personalize dashboard data, ensuring each team member sees relevant information.
52. **What widget can track the deployment success rate across environments?**
    A. Deployment Status
    B. Pipeline Summary
    C. Test Plan Coverage
    D. Work Item Progress
    Answer: A
    Explanation: The Deployment Status widget displays deployment success rates and status across environments, providing insights into CI/CD health.
53. **How can you display detailed error logs from a failing pipeline on the dashboard?**
    A. Use a Query Results widget.
    B. Add the Pipeline Logs widget.
    C. Enable advanced logging in the pipeline settings.
    D. Manually copy logs to a text widget.
    Answer: B
    Explanation: The Pipeline Logs widget surfaces detailed error logs directly on the dashboard, improving visibility for troubleshooting.
54. **You want to display the top contributors to a repository on a dashboard. Which widget should you use?**
    A. Code Contributors
    B. Work Item Progress
    C. Pull Request Summary
    D. Test Plan Trends
    Answer: A
    Explanation: The Code Contributors widget shows statistics about top contributors, commits, and repository activity.
55. **How can you track sprint burndown metrics on the Azure DevOps Dashboard?**
    A. Add the Burndown Chart widget.
    B. Use the Velocity Chart widget.

C. Integrate with Azure Monitor.
D. Add a Query Results widget.
Answer: A
Explanation: The Burndown Chart widget visualizes the remaining work in a sprint, helping teams track progress toward sprint goals.