# **CheetSheet**

## **Single Option**

1. Static Code Analysis: **Build**
2. Track user stories and bugs on the Kanban Board, Track the bugs and tasks on the task board: **Agile**
3. Elapsed time from creation of work items to their completion: **Lead Time**
4. Elapsed time to complete work once they are active: **Cycle Time**
5. Remaining work: **Burndown**
6. Completed work: **Burnup**
7. Roll out limited deployment or small subset of Azure Kubernetes Service (AKS): **Canary Deployment**
8. Replace fixed of existing instances: **Rolling Deployment**
9. Deploy and Roll back in the shortest time or transfer user traffic from old version to new version: **Blue/Green Deployment** or **Red/Black Deployment**
10. Replace JIRA: **Azure Boards**
11. Allows cross-team and teams to collaborate and work together on single development tasks or work: **Azure Boards**
12. Replace Jenkins: **Build Pipeline**
13. Replace Octopus: **Release Pipeline**
14. Authentication mechanism to register self-hosted agent: **Personal Access Token (PAT)**
15. Track product backlog items and bugs on the Kanban Board. Break the PBIs down into tasks board: **Scrum**
16. Reduce the number of queued builds and the time it takes to run the builds for long build times: **Buy self-** **hosted parallel jobs**
17. Reduce the number of queued builds and the time it takes to run the builds for 360 minutes: **Buy Microsoft-** **hosted parallel jobs**
18. Update the Azure SQL Database: **DACPAC**
19. Records all the transactions and changes to the database: **LDF File**
20. Contains all the information in a database: **MDF File**
21. Stores in Azure Blob storage or in local storage in an on-premises location: **BACPAC**
22. IIS web application for default agent pool: **Hosted**
23. AI Service: **Computer Vision**
24. Manage Kubernetes service: **helm**
25. Secure delegated access to resources residing in Azure Storage Account or secrets to store the secrets required by the mobile applications: **Shared Access Authorization (SAS) Token**
26. Storing and accessing secret or prevent the user viewing the account credentials and connection strings: **Azure Key Vault**
27. Reduce Image Size: **Multi-Stage Builds**
28. Increase Image Size: **Single-Stage Builds**
29. Orchestral tool: **Docker Swarm**
30. A management platform in PowerShell that enables you to manage your IT and development infrastructure with configuration as a code: **PowerShell Desired State Configuration**
31. Create and manage containers in AKS: **Azure Service Principal**
32. Containers to publish internally: **Azure Container Registry**
33. Automate containers image builds: **Create and run Azure Container Registry task**
34. Dependabot to apply the update: **Approve the pull request**
35. Communication to isolate the members of different project teams and isolate the members of different project teams: **Microsoft Teams**
36. Visualize the flow: **Kanban Board**
37. Authenticate for granting Azure pipeline accesses to GitHub or runs under the Azure Pipelines identity: **GitHub App**
38. Personal GitHub identity but does not support GitHub: **OAuth**
39. Personal GitHub identity but does not support GitHub: **PAT**
40. Detect known exploits in the Docker images: **Task executed in the continuous integration pipeline and a scheduled task which analyzed the image registry**
41. Prevent the results from being published to the pipeline: **Multi-Stage docker file**
42. Minimize the cost of two build pipelines or minimize the possibility that the two build pipelines will conflict with each other: **Create two container jobs**
43. Distribution groups to control access for application developers who are invited by email: **Private**
44. Distribution groups to control access for early release users who use unauthenticated public links: **Public**
45. Distribution groups to control access for application developers of all apps within the company: **Shared**
46. Only production dependencies are scanned: **npm install**
47. Publish a package to the registry so that it can be installed by name again: **npm publish**
48. Edit an installed package: **npm edit**
49. Work items tracking or bug tracking: **Jira**
50. Override the history of Git repository: **Rebase**
51. Integrate changes from another branch or update the main branch to capture the changes made and override history: **Merge**
52. Fetching a particular branch: **Pull**
53. Keep track of our progress and changes: **Comment**
54. Version control tools to manage our code or store source code: **Azure Repository**
55. Centralized version control or project leads must be able to restrict access to individual file and folders in the repository: **Team Foundation Version Control**
56. Automatically builds and tests code projects to make them available to others or store automated build and test logs: **Azure Pipelines**
57. Queries Language in Azure Logs and Azure Application Insights: **Kusto Query Language**
58. Continuously build, test, release and monitor your mobile and desktop apps: **Microsoft Visual Studio App Center**
59. Multiple latest builds pending deployment: **Deployment Queue Settings**
60. Automatically collect health signals from external service: **Release Gates**
61. Monitor the events for the pipelines: **Azure Pipelines with Slack**
62. Incremental builds without purging or deploy artifacts: **Self-hosted agent**
63. Single feed which can store packages or release of packages that are in development is restricted: **Upstream Source**
64. Applications are build using Erlang: **Azure DevOps self-hosted agents on virtual machines which run on Azure Stack**
65. Limit the release of packages: **Feed Views**
66. To deploy an application: **Deployment**
67. Test servers remain correctly configured: **Set the -ConfigurationMode parameter to ApplyAndAutocorrect**
68. Email alert mail alert is generated whenever virtual machine scale set scales in or out: **Create an action group**
69. Prevent license violation and prohibited libraries or automatically build runs: **Continuous Integration**
70. Send a notification to Jenkins or send build notifications: **Create a service hook subscription**
71. Reduce the history volume or branch uses only a portion of the code in the pull request: **Cherry-pick**
72. Reduce the pipeline execution time: **Pipeline caching**
73. Vulnerability assessments: **Virtual Machines**
74. Virtual machine images that use for the Microsoft-hosted agent pool: **Windows VM**
75. Create an alert based on the page load experience of most users, the alerting level must be based on: **Percentile\_duration\_95**
76. Create an alert when authentication error occurs on the server, the query must be filtered on: **Success**
77. Developer’s access levels to ensure that developers can request and gather feedback from the pilot users: **Basic**
78. Pilot user’s access levels to ensure that developers can request and gather feedback from the pilot users: **Stakeholder**
79. Feature usage: **User Flows**
80. Number of people who used the actions and its features: **Users**
81. Effect that the performance of the application has on the usage of a page or a feature: **Impact**
82. Aggregation granularity (Period): **5 minutes**
83. Threshold value: **Static**
84. Operator: **Greater than**
85. Track requirements, change requests, risks and reviews: **CMMI**
86. Permission type for least privilege: **Secret**
87. Access method for least privilege or restrict access to delete the key vault: **Azure Key Access policy**
88. Enable key vaults for template deployment by using: **A Key Vault advanced**
89. Organization’s role for the Azure DevOps organization: **Reader**
90. Project’s role for the Azure DevOps organization: **Service Account**
91. Service connection type for accessing Azure Key Vault secrets: **Team Foundation Server/Azure Pipelines Service Connection**
92. Authentication/Authorization method for the connection to access Azure Key Vault secrets: **Managed Service Identity Authentication**
93. Groups to control build access: **Microsoft Visual Studio App Center distribution groups**
94. Create a code wiki: **Project Administrators**
95. Object to create: **Build Task**
96. Pull request: **Threat Modeling**
97. Security tool for Continuous Integration: **Static Code Analysis**
98. Continuous Delivery: **Penetration Testing**
99. Convert the .docx files: **Markdown (.md)**
100. Convert the flow charts: **Mermaid Diagrams**
101. Registry information to connect to package repository: **Project.json file**
102. Credentials to connect to package repository: **.npmrc file in user’s home folder**
103. Storage location to store the secrets required by the mobile applications: **Azure storage with HTTP access**
104. Committed code must compile successfully: **A check-in policy**
105. Pull requests must a Quality Gate status of passed in SonarCloud: B**uild Policy**
106. 10 VMs hosted in Azure: **Resource Group**
107. 10 VMs hosted in an on-premises data center environment: **Deployment Group**
108. Releases will be made available to users who are grouped by the tolerance for software faults: **Progressive Exposure**
109. Code will be deployed to enable functionality that will be available in later release of the app: **Feature Flags**
110. To migrate to Azure DevOps on the TFS Server: **Upgrade TFS to the most recent RTM release**
111. To migrate to Azure DevOps perform the migration: **Use the TFS database import service**
112. An application that runs on iOs: **Hosted macOs**
113. Configure the release retention policy for global release: **Set the default retention policy to 30 days**
114. Configure the release retention policy for production stage: **Set the stage retention policy to 60 days**
115. Restrict access to the secrets in Key Vault by using: **RBAC**
116. Version Control must review by at least two project member and associated to at least one work teams: **Git**
117. Project member must be able to work in Azure Repos directly from Xcode: **Subversion**
118. Release branch must only be viewable or editable by the project leads: **Perforce**
119. Use files stored on an SMB-based share from the container’s file system: **kubernetes.io/azure-file**
120. Use files on a managed disk from the container’s file system: **kubernetes.io/azure-disk**
121. Security access x-509 certificates from the container’s file system: **azurekeyvault-flexvolume**
122. Build agent: **Source control system**
123. Release agent: **Hosted service**
124. Release command in GitHub: **git tag –a v3.0 –m “Release v3.0”**
125. Branch type to implement GitFlow workflow strategy for production code: **Master**
126. Branch type to implement GitFlow workflow strategy for pre-production code: **Develop**
127. Source Code: **Azure Repos**
128. Common library used by multiple applications: **Azure Artifacts**
129. Logs from automated test and builds: **Azure Pipelines**
130. Large and frequently updated binary assets: **Azure Storage**
131. Quality and collaboration: **Azure Test Plans**
132. Preconfigured YAML file to manage future pipeline configuration changes: **Git in Azure Repos**
133. Version control system for source code to host on managed Windows server on company’s network: **GitHub Enterprise**
134. Centralized version control system to ensure developers work with most recent version: **BitBucket Cloud**
135. Component to modify continuous delivery: **Internal Review Stage**
136. Can’t be automated: License Procurement
137. Accessed from devices connected from company’s on-premises network or use multi factor authentication to access Azure application from untrusted network: Configure conditional access
138. Performance baseline are prevented from being deployed: Use gate
139. Scan project for common security weaknesses in the open source libraries: **Create a build task and use the WhiteSource Bolt service**
140. Navigate to Compute and Apps: **Azure Security Center**
141. User feedback on the functionality of new apps is received: **Microsoft Test & Feedback extension**
142. Alert when the web app is unavailable from specific Azure regions: **Create Azure Application Insights availability test and alert**
143. Alert is sent when the web app has a sudden rise in performance issues and failures or detect an abnormal rise in the rate of failed: **Smart Detection**
144. Display the commit status of the repository on Azure Boards: **Add the Azure Boards app to the repository**
145. Receive events about failed builds in Microsoft Teams: **From Microsoft Teams, run @azure pipelines subscriptions**
146. Receive Microsoft Teams notifications when work items are updated: **Configure a connector**
147. Action to alert rule in Azure Monitor: **Failed attempt to delete the ASP-9bb7 resource**
148. Generate alert or capture and store telemetry data: **Log Analytics**
149. Trigger to invoke the logic app: **Request Trigger**
150. Restart Azure Container Instance when App stops responding: **Add a liveness probe to the YAML configuration**
151. Create the consolidated view: **Azure Dashboards**
152. Automate UI testing of a web application: **Selenium**
153. Implement health checkup: **Application Health Extension**
154. Configuration management solution: **Azure App Configuration**
155. Handle incoming request before users can submit request to application: **Readiness Probe**
156. Authenticate users using Azure Active Directory: **Create An App Registration In Azure AD**
157. Azure feature for security of application: **Compute And Apps In Azure Security Center**
158. Minimize infrastructure credentials to be leaked: **Add Azure Key Vault Reference To Azure Resource Manager Templates**
159. Key vault access policy to assign secret permission to application: **Gate Only**
160. Two refer Azure key vault secrets in a build pipeline: **Create A Variable Group**
161. Effect of policy in tenant route group: **Ensures That All Traffic To New Azure Storage Account Is Encrypted**
162. Configure GitHub to use Azure Active Directory: **Register Github In Azure AD**
163. Identify package dependency: **Sonarqube**
164. Unlist and depreciate packages or custom work item queries to report on the project's progress or Edit wiki pages: **Contributor**
165. Certificate file to distribute new release from app center: **. P12**
166. Restore the NuGet package automatically: **Azure Artifacts Credential Provider**
167. Configure organization setting for Azure pipeline: **Third Party Application Access Via Oauth**
168. Continuous inspection of the company’s code base: **Sonarcloud Analysis**
169. Improve quality of code: **Run PDM**
170. Register-AzureRmAutomationDscNode: **Change the value of the ConfigurationMode parameter**
171. Communication between members: **Slack or Microsoft Teams**
172. Distribute a new iOS application to the distribution group by using Microsoft Visual Studio App Center: **Register the devices on the Apple Developer portal**
173. Enable the service in App1 once App2 is deployed: **Implement a feature flag**
174. Release information for the pipeline is added automatically to the work items: **Modify the post-deployment conditions for the last stage of the pipeline**
175. Release pipeline that will deploy resources by using Azure Resource Manager templates: **Azure PowerShell**
176. Branching strategy for the team’s Git repository: **Single-running branch with multiple short-lived topic branches**
177. Investment planning applications suite: **Development Isolation**
178. Compute type for the agent: **Azure virtual machine scale set**
179. Create a staging environment: **configure new tasks in the release pipeline to create and delete the virtual machines m Azure DevTest Labs**
180. Collect crash reports, distribute bash releases, get users feedback: **Microsoft Visual Studio App Center integration**
181. Erlang and Hack: **AzureDevOps self-hosted agents on virtual machine that run on Azure Stack**
182. Merge change to the master branch: **modify the access control for the user**
183. Prevent from viewing account credentials and connection strings: **Azure Key Vault**
184. enforce a pull request: **Squash Merge**
185. generate a ticket in Service Now when the application fails to authenticate: **It Service Management Connector (ITSM)**
186. clone the repository from GitHub to Azure DevOps: **From Import A Git Repository, Click Import**
187. Azure Artifacts for Python packages: **PyPpi**
188. change management to protect master branch, review change, reflect changes in master branch: **Change Management**
189. Minimize the number of queued builds and the time it takes to run the builds: **Purchase Self-Hosted Parallel Jobs**
190. create Project3: **Create A Service Endpoint**
191. integrate image scanning: **Task Executed In The Continuous Integration Pipeline And A Scheduled Task That Analyzes The Image Registry**
192. add the virtual machines as managed nodes in Azure Automation State Configuration: **Run The Register-Azurermautomationdscnode Azure Powershell Cmdlet**
193. Integrate GitHub, Provide integrated debugging tools, support remote workers and developers: **Visual Studio Codespaces**
194. single feed that stores packages produced: views in Azure Artifacts
195. build pipeline completes successfully if the third-party applications are unavailable: **Configure Flaky Tests**
196. Image is updated automatically whenever the base image is updated: **Create And Run An Azure Container Registry Task**
197. Implement Project4: **Add A Docker Task To The Build Pipeline**
198. To reduce image size: **From Microsoft/Dotnet:2.1-Sdk**
199. Implement the code quality restriction: pre deployment approval
200. What prevents the Smart Detection notification from being sent: **Smart Detection Uses The First 24 Hours To Establish The Normal Behavior Of The Web App**

## **Multiple Options**

1. Java: **Gradle, Maven**
2. Technical debt: **Time spent on rework, reducing code coupling and the dependency cycle, reduce the code complexity**
3. Strategy for managing technical debt: **Integrate Azure DevOps and SonarQube, Configure post-deployment approvals in the deployment pipeline.**
4. Security: **Azure Sentinel, Azure Firewall**
5. Release pipeline to deploy resources using ARM Template: **Use Azure Powershell, Add multiple Azure DevOps services build steps, each one performing a stage task**
6. Open Source Libraries: **Whitesource Bolt, Black Duck**
7. Java-Based application to add code coverage: **JaCoCo, Cobertura**
8. Reduce the time of build pipeline: **Use Agent Pool, Increase number of parallel jobs**
9. Retrospective for scrum master: **What went well in Sprint?, What could be improved?, What will be commit to improve in the next Sprint?**
10. Automatically send daily summary of the exceptions: **Logic Apps, Application Insights**
11. Implement Azure DevOps to control the build process and the build server using self-hosted agent: **Create PAT from DevOps, Run config.cmd from the build server**
12. Store huge frequently updated binary assets: **Azure Storage, Git Large File Storage**
13. Prevent configuration: **Continuous Assurance, Subscription Health Scan**
14. Deployments only fail if the approvals take longer than eight hours: **From Pre-deployment conditions, you modify the Time between reevaluation of gates option.**
15. Integration strategy for the build process of a Java application: **Configure an Octopus Tentacle, Configure the build pipeline to use a Hosted Ubuntu agent pool**
16. Deploy Azure resources by using Azure Resource Manager templates: **Main template that has two linked templates, Two standalone templates, each of which will deploy the resources in its respective group**
17. Maintain a uniform configuration: **Azure Resource Manager templates, The Custom Script Extension for Windows**
18. Increase the logging level: **An Azure Automation runbook, An Azure Monitor alert that has a dynamic threshold**
19. Pinpoint the average load times of the application pages: **Azure Application Insights, Azure Advisor**
20. Send SMS alert: **Create an Azure Service Health alert, Create and configure an action group**
21. Capture the telemetry data: **Add Azure Application Insights telemetry, Disable adaptive sampling**
22. Prevent the build pipeline from failing due to the test: **Clear Flaky tests included in test pass percentage, Manually mark the test as flaky**
23. Collect detailed data: **Azure Log Analytics agent, Dependency Agent**
24. WhiteSource Bolt only scans production dependencies: **Run npm install and specify the --production flag, Modify the devDependencies section of the project's Package.json file.**
25. Minimize the number of publication points: **Create a new feed for the** **package, Publish the package to a public NuGet repository.**
26. Delete the file and its history: **the git filter-branch command, BFG Repo-Cleaner**
27. Automate the assignment of the code reviews: **Select Never assign certain team members, Set Routing algorithm to Load balance.**
28. Manage the build and release processes: **An External Git service connection, a self- hosted agent**
29. Change request is processed before any components can be deployed: **Define a deployment control that invokes the Service Now SOAP API, Define a post deployment gate after the deployment to the QA stage.**
30. Test the application manually in iOs device: **Deploy a certificate from an internal certification authority (CA) to each device, Register the IDs of the devices in the Apple Developer portal.**
31. Email message when the front end fails: **availability tests in Azure Application Insights, Application Map in Azure Application Insights**
32. Terraform to deploy an Azure resource group: **Terratest, Yeoman**
33. Notifications in Microsoft Teams: **Create an Azure logic app that has an HTTP request trigger, Modify the Diagnostics settings in Azure Monitor.**
34. Values to authenticate using Azure AD: **Tenant ID, App ID, Client secret**
35. **Implement limited user testing for the new version: Upload a custom script file to sa1, Modify the Custom Script extension settings of VMSS1, Update the configuration of a virtual machine in VMSS1.**
36. Secure API: **Modify the application to access the key vault, Configure a Key Vault access policy, Deploy a virtual machine that uses a system-assigned managed identity.**
37. Custom package managed centrally and the latest version: **Publish the package to a feed, Create a new feed in Azure Artifacts, Add the package URL to the NuGet Package Manager settings in Visual Studio.**
38. Jenkins can retrieve source code from Azure Repos: **Add the Team Foundation Server (TFS) plug-in to Jenkins, Create a personal access token in your Azure DevOps account, Create a service hook in Azure DevOps.**
39. Integrate Azure DevOps and BitBucket: **Self-hosted agent, external git service**
40. Summary of the exceptions sent to Microsoft Teams: **Azure Logic Apps, Azure Application Insights**
41. Configure a probe to confirm pod is responding, check status of pod, initiate a shutdown: **redinessprobe, periodsecond:15**
42. Identify distinct event ID: **summarize, makelist(EventID)**
43. Create a report for failed webapp: **Requests, success==false**
44. Developers receive Microsoft Teams notifications when there are failures in a pipeline: **subscribe** [**https://dev.azure.com/contoso/contoso-app**](https://dev.azure.com/contoso/contoso-app)
45. To create a test in Azure Application Insights the test will execute: **Every 5 minutes at a random location**
46. To create a test in Azure Application Insights the test will pass within 30 seconds: **All the HTML, JavaScript and images of App1 load**
47. Implement a policy to meet the tagging requirement: **“MicrosoftResources/subscriptions/resourceGroups”, “Deny”**
48. Initialize App Center in mobile application: **MSAnalytics.self, MSCrashes.self**
49. Configure filters: path filter to exclude, branch filter to include
50. Dynamically generate the resource ID of the key vault during template deployment: **"Microsoft.Resources/deployment”, “templateLink”**
51. DSQ configuration script: Configuration, WindowsFeature

## **Arrangement**

1. Helm and Tiller: **kubectl create > helm init > helm install**
2. Associate an automated test to a test case: **Create test project > Check in the project onto DevOps Repo > Add automated test to pipeline**
3. Docker Trusted Registry: **az aks create > az ad sp create-for-rbac > kebectl create**
4. Configure alert to forward details of error: **Create an Azure logic app > Select the HTTP request trigger > Update the action group in Azure Monitor**
5. Set AKS cluster by using RBAC roles: **Cluster > System-assigned managed identity > RBAC binding**
6. Link GitHub commits, pull requests, and issues to the work items: **Register a new OAuth app > Create a service hook subscription > Add a GitHub connection**
7. Use the secret in the deployment pipeline: **Create a service principal in Azure AD > Configure an access policy in the key vault > Add an Azure Resource Manager service connection to the pipeline**
8. Azure Container Instances for running the containers in AKS: **YAML file > Run kubectl > helm init**
9. Deploy code using pipeline: **Create service principal > Create service connection > Configure permission**
10. Configure SonarQube: **Prepare Analysis Configurayion > Virtual Studio Build > Virtual Studio Test > Run Code Analysis**
11. Find and isolate shared code: **Create a dependency graph > Group the related components > Assign ownership**
12. Share and test package: **Configure self-hosted agent > Create a new Azure Artifacts Feed > Publish package > Connect to Azure Artifacts Feed**
13. Configure Azure Automation: **Create a DSC configuration file that has an extension of .ps1 > Run Import-AzureRmAutomationDscConfiguration > Run Start-AzureRmAutomationDscCompilationJob**
14. Consolidate package into single feed: **Create NuGet package > Create Azure Artifacts > Create Microsoft Visual Studio project**
15. Deploy resources using ARM template: **Create a release pipeline > Add an Azure Resource Group Deployment task > Set the template parameters**
16. VM to have same windows feature installed always: **Create a PoweShell configuration file > Load the file > Configure Custom Script Extension**
17. Prevent config.json from being committed: **Delete and recreate repository > Add Config.json > Run git add .gitignore**
18. Manage ongoing consistency: **Assign node configuration > Upload configuration > Compile configuration > Onboard VM > Check compliance status**
19. Implement code flow strategy: **Create repository > Create branch > Add a build validation policy**
20. Implement project: **Add a manual intervention task > Add Query > Enable gates**
21. Onboard Linux VM: **Create DSC metaconfiguration > Copy metaconfiguration > Add VM**
22. Deploy application to VM: **Create agent pool > Create deployment group > Add Azure pipelines Agent > Add and configure deployment group job**
23. Complete the pipeline to configure OWASP ZAP: **Call Baseline Scan > Download file > Convert Report Format > Publish Test Results > Destroy OWASP Container**
24. Create Dockerfile: **From Microsoft/dotnet:2.2-sdk > Run dotnet restore > From Microsoft.com/dotnet/2.2-aspnetcore-runtime**

## **Note**

1. **Build vm has just one job, Initialize Build**
2. **Stages of a pipeline are: Build vm, Deploy\_to\_dev, Deploy\_to\_uat, Finalize Build**
3. **There are five stages: Development, QA, Pre-production, Load Test and Production. They all have triggers.**
4. **Cloud Agent is job**
5. **Cloud Agent have 4 Tasks: Nuget Restore, Compile Application, Copy Files, Publish Artifact**
6. **OAuth, PAT does not work with Azure Pipeline**
7. **Dependabot by default create a pull request and create a branch**