W13 Fast Focus – Azure Pipelines Tips and Tricks – Demo Script v1.0

|  |  |  |
| --- | --- | --- |
| Topic | Talking Points | Demo Script |
| What Is Azure Pipelines | * Azure Pipelines is a cloud service that you can use to automatically build and test your code project and make it available to other users. It works with just about any language or project type. * Azure Pipelines combines continuous integration (CI) and continuous delivery (CD) to constantly and consistently test and build your code and ship it to any target. * Azure Pipelines integrates with GitHub, GitHub Enterprise, Azure Repos Git & TFVC, Bitbucket Cloud, and Subversion. * Use Azure Pipelines to deploy your code to multiple targets. Targets include container registries, virtual machines, Azure services, or any on-premises or cloud target. * Build Agents      * Triggering Builds   + CI trigger   + Manual Trigger   + PR Trigger   + Scheduled Trigger   + Pipeline Triggers      * MORE INFO   + <https://docs.microsoft.com/en-us/azure/devops/pipelines/get-started/what-is-azure-pipelines?view=azure-devops> | Make this a slide I go through when discussing this point. |
|  |  |  |
| Pipeline Types - Classic vs Multi-Stage YAML | * CLASSIC   + GUI Interface   + Build Pipelines   + Release Pipelines   + Currently has most features * YAML   + Multi-stage vs Build only   + .yml file   + Pipeline is versioned with code   + Branches can modify build policy by modifying .yml file in branch * MORE INFO   + <https://docs.microsoft.com/en-us/azure/devops/pipelines/get-started/pipelines-get-started?view=azure-devops> | * TEAM PROJECT: <https://dev.azure.com/devopsmickey/LAB-AzureKubernetesService> * PREP   + Make sure VSS is up and running * Open MyHealth.AKS.build and Edit Build to show tasks. Discuss * Open MyHealth.AKS.Release and Edit Release to show release pipeline. Discuss * Open AKS YAML pipeline, which does the same as the other two. Discuss |
|  |  |  |
| Variable Groups / Tasks Groups | * Variable Groups   + Store values that you want to control and make available across multiple pipelines   + Store secrets and other values to be passed to YAML pipelines   + Defined/Managed in Pipelines | Library   + Can't be created in YAML pipeline, but can be used * Task Groups   + Classic only. For YAML, you should use template files   + A task group allows you to encapsulate a sequence of tasks, already defined in a build or a release pipeline, into a single reusable task that can be added to a build or release pipeline, just like any other task. You can choose to extract the parameters from the encapsulated tasks as configuration variables, and abstract the rest of the task information. * MORE INFO   + <https://docs.microsoft.com/en-us/azure/devops/pipelines/library/variable-groups?view=azure-devops&tabs=yaml>   + <https://docs.microsoft.com/en-us/azure/devops/pipelines/library/task-groups?view=azure-devops> | * TEAM PROJECT: <https://dev.azure.com/devopsmickey/LAB-AzureKubernetesService> * Open the MyHealth.AKS.build-UsingGroups pipeline * Go to variables * Show the variable group * Show how to edit the variable group * Go to the tasks * Show how you created the task group * Show how to edit the task group * Work for build and release |
|  |  |  |
| Build Logs and System.Debug | * Everything that happens in the build is logged * You can turn on more verbose logging to help with debugging | * Open the MyHealth.AKS.build/#20200221.2 Build Report * Duration, Commits, Artifacts * Select Job and view Build Report * Open the MyHealth.AKS.build/#20200221.3 to see what logs look like with Debug turned on (specifically compare the second replace tokens to see) |
|  |  |  |
| Disable Tasks To Help Debug Issues | * Sometimes you don't want to run every task, you want to run step by step * In YAML, you can comment out task * In Classic Pipelines you can just disable unwanted tasks | * Open MyHealth.AKS.build and show how you can disable 1+ tasks * Open YAML and show how you could comment things out * Mention you could also use VS Code to edit this (it is just a file) and it gives intellisense. You also get intellisense here, and the task picker on the side |
|  |  |  |
| Tie Pipelines Back To Work Items | * You can associate code changes with a work item * When the build pipeline runs, you can see what work items are satisfied with the build * With Classic Pipelines, when the release pipeline runs, you can see what current "stage" the work item is in | * When you make code changes, you can associate your code change with a work item. That is what I did (2669, 2670) * Open MyHealth.AKS.build/#20200221.2 and you can see the work items that were part of the build. * Open MyHealth.AKS.release/Release-4, select DEV stage, and show results, and how it ties back to work items * Edit the pipeline, go to options, and show how you can set stage names * Open the work items and show what you see |
|  |  |  |
| Tokens/Variable Replacement | * You can add tokens to files, and replace those tokens with pipeline variables during the build/deploy process * For web.config, you can use transform/substitution build into App Service Deploy task * Multiple marketplace token replacers available. Replace token from Colin * Token replace in build vs release. Usually release, to make config changes for a specific environment | * Open the appsettings.json file in the src/MyHealth.web * See the tokens * Open the build and look at the replace tokens task * You can also do this in the release as well * Add an app service deploy task, and talk about how the web.configs could be transformed |
|  |  |  |
| Branch Policies - Build as part of PR Verification | * Guarantee changes build before they get to master * Can do other things as well (reviews, etc) * For GIT * MORE INFO * <https://docs.microsoft.com/en-us/azure/devops/repos/git/branch-policies-overview?view=azure-devops> | * TEAM PROJECT: <https://dev.azure.com/devopsmickey/LAB-AzureKubernetesService> * Open Projects Settings | Repositories * Select Master Branch and Select Policies * You can see the different policies and the build I've scheduled whenever a pull request is opened against master * You can have multiple builds * Go to repo and modify readme.md on the dev branch. * Save it, and because I have an active PR, it is added to that, and you can see the build is kicked off |
|  |  |  |
| Environments | * Collection of resources that can be targeted by a deployment from a pipeline * Provides consolidated information such as: Deployment History, Traceability, Permissions * We are going to look at it from an AKS perspective * MORE INFO * <https://docs.microsoft.com/en-us/azure/devops/pipelines/process/environments?view=azure-devops> | * TEAM PROJECT: <https://dev.azure.com/devopsmickey/LAB-AzureKubernetesService> * Edit the YAML pipeline * Show the environment tag under deployment. This ties it back to an environment called Dev * Start creating a new kubernetes env to show the process * Click into DEV env * Click Deployments to see deploys * Click resources and select default * Click services to see IP addresses |
|  |  |  |
| Approval and Gates | * Control over start/end of each stage in a pipeline * Manual Approvals * Automated System Checks * MORE INFO * <https://docs.microsoft.com/en-us/azure/devops/pipelines/release/approvals/?view=azure-devops> | * TEAM PROJECT: <https://dev.azure.com/devopsmickey/LAB-AzureKubernetesService> * Open Releases | MyHealth.AKS.Release * Clone DEV - talk about how it clones everything. Call it QA * Look at post-deployment conditions for DEV * Look at the pre-deployment conditions for QA |