Bluebird YAML Pipelines

DevOps

The goal of DevOps is to shorten the systems development life cycle while also delivering features, fixes, and updates frequently in close alignment with business objectives. The DevOps approach is to include automation and event monitoring at all steps of the software build.

~ Wikipedia Circa 2018

In order to complete this idealology we must first have pipelines defined within our code. This will allow us to update the development pipeline while making development changes.

Background

The ability to write pipelines as code within Azure devops is only partially complete. Build pipelines are able to be defined in code however release pipelines as code is not yet a reality (as of 12/10/2018).

Release Schema

The release definition schema is defined by the REST API that Microsoft has recently exposed to the public. This schema can be used to construct a pipeline programmatically.

For more see the Microsoft Documentation

YAML Abstraction Layer

The JSON schema contains a lot of repeated information and is not necessarily as productive to construct. This brought about the creation of a YAML Abstraction layer

pipelineGenerator.py

To install pre-requisites run this command

```
pip install requests Jinja2 pyyaml click
```

To use the pipeline generator simply run this command:

```
python pipelineGenerator.py pipeline.yml -d
```

The d switch is to trigger the release pipeline after it is updated.

Variables

Variables can be added to the Release definition using the root `variables` and listing every key/value pair underneath.

```yaml
variables:
  variableOne: test
  variableTwo: testAgain
```

Or can be added as a link to a seperate file.

```yaml
variables: variableGroup.yml
```

Artifacts

Build

PropertyName	Type	Description
id	integer	build definition Id
branch	string	branch that the build should be triggered on
name	string	Name of Build Artifact
alias	string	Alias used in the pipeline

```yaml
artifacts:
  - type: build
    id: "135"
    name: "IdentityServer"
    alias: "IDS Alias"
```

Git

PropertyName	Type	Description
id	guid	git repo Identifier
name	string	Name of Repo Artifact
alias	string	Alias used in the pipeline

```yaml
artifacts:
  - type: git
    id: "e8aeff64-c356-4c36-aae4-94dda4159542"
    branch: "master"
    name: "CI-CD"
    alias: "IOT-CI-CD"
```

Artifacts will be brought in under the `$(System.DefaultWorkingDirectory)/{alias}` directory.

Stages

Stages consist of Phases and are logical grouping of Tasks

```
stages:
 - name: "stage 1"
   phases:
```

Phases

Phases are groups of Tasks that are run on a specific agent

```
stages:
 - name: "stage 1"
   phases:
   - name: "Agent Job 1"
     tasks:
```

Tasks

Tasks are specific tasks preformed on the agents. These all require different set of inputs. They are distinguished by the type variable

```
stages:
 - name: "stage 1"
   phases:
   - name: "Agent Job 1"
     tasks:
     - type: "bash"
       name: "Andrew's Bash Script"
       inline: true
       script: "echo 'IOT is cool' | awk '{print \"Team Bluebird \" $2\" \"$3}'"
       workingDirectory: ""
```

These tasks are populating Jinja2 templates. These templates are formed like so:

```
{
    "environment": {},
    "taskId": "6c731c3c-3c68-459a-a5c9-bde6e6595b5b",
    "version": "3.*",
    "name": "{{name}}",
    "refName": "",
    "enabled": true,
    "alwaysRun": false,
    "continueOnError": false,
    "timeoutInMinutes": 0,
    "definitionType": "task",
    "overrideInputs": {},
    "condition": "succeeded()",
    "inputs": {
        "targetType": {% if inline %}"inline"{% else %}"filePath"{% endif %},
        "filePath": "{% if inline %}{% else %}{{filename}}{% endif %}",
```

```
        "arguments": "{% if arguments %}{% for key,value in arguments.items() %}--{{
 key }} {{value | replace('"','\"')}} {% endfor %}{% endif %}",
        "script": "{% if inline %}{{script|replace('\"','\\\"')}}{% else %}{% endif
 %}",
        "workingDirectory": "",
        "failOnStderr": "false"
    }
}
```

Bash

Bash runs a Bash Script either from an inline variable or from a file in the working area.

PropertyName	Type	Description
type	string	"bash"
name	string	Name of the Task (arbitrary)
inline	boolean	determines whether the script is run from file or inline
script	string	Inline script to run (required if inline)
filename	string	filename to run if not inline (required is not inline)
workingDirectory	string	path to working directory
arguments	object	object of key/value pairs for each argument to pass to shell script

Powershell

Runs a powershell script either inline or from a file

PropertyName	Type	Description
type	string	"powershell"
name	string	Name of the Task (arbitrary)
inline	boolean	determines whether the script is run from file or inline
script	string	Inline script to run (required if inline)
filename	string	filename to run if not inline (required is not inline)
workingDirectory	string	path to working directory
arguments	object	object of key/value pairs for each argument to pass to shell script

Azure CLI

Runs a Azure CLI script either inline or from a file. This is likely done within BASH in the backend.

PropertyName	Type	Description
type	string	"cli"
name	string	Name of the Task (arbitrary)
inline	boolean	determines whether the script is run from file or inline
script	string	Inline script to run (required if inline)
fileName	string	filename to run if not inline (required is not inline)
workingDirectory	string	path to working directory
arguments	object	object of key/value pairs for each argument to pass to shell script

Azure SQL

Runs a Azure SQL Script either inline or from file.

PropertyName	Type	Description
type	string	"azuresql"
name	string	Name of the Task (arbitrary)
inline	boolean	determines whether the script is run from file or inline
script	string	Inline script to run (required if inline)
fileName	string	filename to run if not inline (required is not inline)
server	string	server hostname
username	string	sql username
password	string	sql password
database	string	database name

ARM Deployments

Executes an ARM deployment from ARM Template.

PropertyName	Type	Description
type	string	"arm"
armResourceName	string	Name of Resource
resourceGroup	string	resource group to deploy into (create if not existing)
resourceGroupLocation	string	resource group location
resourceArmTemplateFile	string	path to template file
resourceArmParameterFile	string	path to parameter file
overrideParams	object	object of key/value pairs for each parameter to override

App Service

Executes an ARM deployment from ARM Template.

PropertyName	Type	Description
type	string	"appservice"
webAppName	string	Name of web app resource
deployToSlot	object	See below
packagePath	string	path to package (*.zip)
appSettings	object	object of key/value pairs for each application setting

deployToSlot

PropertyName	Type	Description
resourceGroup	string	resource Group for the slot
slotName	string	name of slot

Swap Slots

Exchange slots on Web app service

PropertyName	Type	Description
type	string	"appservice"
webAppName	string	Name of web app resource
resourceGroup	string	Resource Group
source	string	source slot
target	string	target slot (default: production)

Sample YAML

```
name: "Radha Release Pipeline -- delete me"

variables: "crsliotbbdemo.yml"

artifacts:
  - type: build
    id: "135"
    branch: "master"
    name: "IdentityServer"
    alias: "IDS Alias"
  - type: build
    id: "30"
    branch: "master"
    name: "ARM-Templates-import"
    alias: "ARM-Templates"
  - type: git
    id: "e8aeff64-c356-4c36-aae4-94dda4159542"
    branch: "master"
    name: "CI-CD"
    alias: "IOT-CI-CD"
stages:
 - name: "stage 1"
   phases:
   - name: "Agent Job 1 -- Andrew"
     tasks:
     - type: "bash"
       name: "Andrew's Bash Script"
       inline: true
       script: "echo 'IOT is cool' | awk '{print \"Team Bluebird \" $2\" \"$3}'"
       workingDirectory: ""
     - type: "powershell"
       name: "Andrew's Powershell Script"
       inline: true
       script: "echo \"IOT is cool\""
     - type: "azuresql"
       server: "$(ApplicationAcronym)-$(databaseRegion)-
sql-$(deployEnvironment)-$(deployNumber).database.windows.net"
```

```
        inline: true
        script: "SELECT * FROM Product WHERE ProductId=6"
        filename: "$(System.DefaultWorkingDirectory)/ARM-
Templates/drop/Database/IdentityServerTBLScript.sql"
        password: "$(sqlServerAdminUserPassword)"
        username: "$(sqlServerAdminUserName)"
        database: "$(sqlDatabaseName)-$(deployEnvironment)"
      - type: appservice
        name: "Deploy Web App IDS"
        webAppName: "$(foundationName-SWM-Function)"
        packagePath: "$(System.DefaultWorkingDirectory)/SoftwareMgmt-BE-
Dev/drop/Connect3M.SoftwareMgt.AzureFunctions.zip"
        deployToSlot:
         resourceGroup: "$(resourceGroup)"
         slotName: "staging"
        appSettings:
         KeyVaultAppId: "1202a0a1-fe76-452c-a0fa-8506d087d2e2"
         KeyVaultAppKey: "nIspsO4uTv0/R8SnGRQ2HJhfriKSuCmA1nDDLOZmVq8="
         KeyVaultAppUri: "https://crsle2vltdev01.vault.azure.net/"
         TenantId: "facac3c4-e2a5-4257-af76-205c8a821ddb"
         MSDEPLOY_RENAME_LOCKED_FILES: 1
         WEBSITE_HTTPLOGGING_RETENTION_DAYS: 1
         WEBSITE_LOAD_CERTIFICATES: "*"
         ASPNETCORE_DETAILEDERRORS: true
      - type: swapslots
        webAppName: "crsliotisbbdemo"
        source: staging
        resourceGroup: "rg-crsliot-bbdemo"
```