**# AZD Challenges:**

**# - can we hide IaC (eg ARM/Bicep) from the dev but still**

**# allow them to use IaC as part of their gitOps flow?**

**# - can we offer a simplified CLI UX (eg az-like) for azd**

**# when the user wants to create/edit their IaC?**

**# - can we offer an 'escape hatch' to the IaC for the very**

**# advanced use cases where they need to edit IaC manually?**

**# - dealing with edits made outside of local azd/IaC (drift)**

**#**

**# Solution Summary:**

**# - support az-like syntax on azd CLI**

**# - map those requests to edits of local IaC files**

**# - local IaC files only include data user has provided**

**# - local IaC will be augmented with smart/required defaults**

**# prior to being sent to Azure**

**# - local IaC files are based on official Azure IaC, so, when**

**# needed, user can edit them directly for advanced features**

**# without learning (yet another) IaC format**

**# - drift is solved via diff/sync commands (think 'git rebase')**

**# Demo part 1: IaC management w/o seeing IaC directly**

**# - scenario: from scratch, create ACA app + dev-mode Redis**

**# - use a clean dir to show we're not cheating**

**# - PoC uses REST/JSON instead of Bicep just for convenience**

**$** **mkdir testDir1**

**$** **cd testDir1**

**# Init project with rg, sub and loc**

**$** **azx init -g default -s fe108f6a-2bd6-409c-8bfb-8f21dbb7ba0a -l 'East US'**

**# Create a redis dev-mode service (our ACA env 'demo' already exists)**

**$** **azx add aca-redis -n redis1 --environment demo**

**# Create an ACA app from existing image & externalized & bind to redis**

**$** **azx add aca-app -n app1 -i duglin/aca-lister --external --bind redis1**

**# Some things to note**

**# - No IaC exposed during the creation of a new resource**

**# - CLI UX is az-like**

**# - IaC is created on disk, nothing is in Azure yet**

**# If user ever needs access to IaC, they can find it under .azx dir**

**$** **ls .azx/stage\_default**

aca-app-app1.json

aca-redis-redis1.json

**# When 'show'ing a resource, can see 'human' or 'machine' readable**

**$** **azx show aca-app -n app1 # Human readable**

Name : app1

Environment : demo

Location : East US

Subscription : fe108f6a-2bd6-409c-8bfb-8f21dbb7ba0a

ResourceGroup: default

Ingress: external

Containers:

- Image: duglin/aca-lister

Bindings:

- Service: redis1

**$** **azx show aca-app -n app1 -o json # Machine readable**

{

"id": "/subscriptions/fe108f6a-2bd6-409c-8bfb-8f21dbb7ba0a/resourceGroups/default/providers/Microsoft.App/containerApps/app1",

"location": "East US",

"properties": {

"environmentId": "demo",

"configuration": {

"ingress": {

"external": true

}

},

"template": {

"containers": [

{

"image": "duglin/aca-lister"

}

],

"serviceBinds": [

{

"serviceId": "redis1"

}

]

}

}

}

**# Notice the only 'values' shown are what the user has provided.**

**# Not cluttered with info they haven't expressed an interest in, yet**

**# Everything is still local, so provision it in Azure now.**

**# Notice redis goes first because the app depends on it**

**$** **azx up**

Provision: aca-redis/redis1

Provision: aca-app/app1

**# When 'show'ing a resource, can see IaC(above), REST (IaC+defaults),**

**# or what's in Azure. In 'human' or 'machine' readable.**

**$** **azx show aca-app -n app1 --from rest # Human readable + defaults**

Name : app1

Environment : /subscriptions/fe108f6a-2bd6-409c-8bfb-8f21dbb7ba0a/resourceGroups/default/providers/Microsoft.App/managedEnvironments/demo

Location : East US

Subscription : fe108f6a-2bd6-409c-8bfb-8f21dbb7ba0a

ResourceGroup: default

Workload Profile Name: Consumption

Ingress: external

Port: 8080

Containers:

- Image : duglin/aca-lister

CPU : 0.5

Memory : 1Gi

Max Scale: 10

Bindings:

- Service: redis1

Name : redis1

**# Notice it has fields that azx filled in automatically for the user**

**# Now look at what's in Azure in JSON instead of human readable**

**$** **azx show aca-app -n app1 --from azure -o json # Machine readable from Azure**

{

"id": "/subscriptions/fe108f6a-2bd6-409c-8bfb-8f21dbb7ba0a/resourceGroups/default/providers/Microsoft.App/containerapps/app1",

"identity": {

"type": "None"

},

"location": "East US",

"name": "app1",

"properties": {

"provisioningState": "Succeeded",

"runningStatus": "Running",

"managedEnvironmentId": "/subscriptions/fe108f6a-2bd6-409c-8bfb-8f21dbb7ba0a/resourceGroups/default/providers/Microsoft.App/managedEnvironments/demo",

"environmentId": "/subscriptions/fe108f6a-2bd6-409c-8bfb-8f21dbb7ba0a/resourceGroups/default/providers/Microsoft.App/managedEnvironments/demo",

"workloadProfileName": "Consumption",

"outboundIpAddresses": [

"20.231.246.122",

"20.231.246.54",

"20.231.247.19",

"20.231.246.253",

"20.241.227.6",

"20.241.226.169",

"20.127.248.50",

"20.241.171.30",

"20.241.172.248",

"20.241.172.250",

"20.246.203.138",

"20.246.203.140",

"20.124.73.117",

"4.156.169.214",

"4.156.169.175",

"4.156.169.143",

"20.241.173.137",

"20.241.173.98",

"20.242.228.13",

"20.242.227.204",

"20.242.227.238",

"20.242.228.93"

],

"latestRevisionName": "app1--meuhy54",

"latestReadyRevisionName": "app1--g6s7808",

"latestRevisionFqdn": "app1--meuhy54.orangetree-90dcdba8.eastus.azurecontainerapps.io",

"customDomainVerificationId": "937A4FAC51188B24CE3789FD7422E2353001D813C26FDFEB4E34BE3C8A645E3E",

"configuration": {

"secrets": null,

"activeRevisionsMode": "Single",

"ingress": {

"fqdn": "app1.orangetree-90dcdba8.eastus.azurecontainerapps.io",

"external": true,

"targetPort": 8080,

"exposedPort": 0,

"transport": "Auto",

"traffic": [

{

"weight": 100,

"latestRevision": true

}

],

"customDomains": null,

"allowInsecure": false,

"ipSecurityRestrictions": null,

"corsPolicy": null,

"clientCertificateMode": null,

"stickySessions": null,

"additionalPortMappings": null

},

"registries": null,

"dapr": null,

"maxInactiveRevisions": null,

"service": null

},

"template": {

"revisionSuffix": "",

"terminationGracePeriodSeconds": null,

"containers": [

{

"image": "duglin/aca-lister",

"name": "main",

"resources": {

"cpu": 0.5,

"memory": "1Gi",

"ephemeralStorage": "2Gi"

}

}

],

"initContainers": null,

"scale": {

"minReplicas": null,

"maxReplicas": 10,

"rules": null

},

"volumes": null,

"serviceBinds": [

{

"serviceId": "redis1",

"name": "redis1"

}

]

},

"eventStreamEndpoint": "https://eastus.azurecontainerapps.dev/subscriptions/fe108f6a-2bd6-409c-8bfb-8f21dbb7ba0a/resourceGroups/default/containerApps/app1/eventstream"

},

"systemData": {

"createdBy": "dougdavis@microsoft.com",

"createdByType": "User",

"createdAt": "2024-01-04T18:35:31.9883625",

"lastModifiedBy": "dougdavis@microsoft.com",

"lastModifiedByType": "User",

"lastModifiedAt": "2024-01-04T18:38:50.8882189"

},

"type": "Microsoft.App/containerApps"

}

**# Demo part 2: Dealing with IaC 'drift'**

**# - Changes are made in Azure that our local IaC doesn't know about**

**# Change something locally : remove the binding**

**$** **azx update aca-app -n app1 --unbind redis1**

**# Change something in Azure : add an environment variable, set max scale**

**$** **az containerapp update -n app1 --set-env-vars abc=def --max-replicas=9**

WARNING: The behavior of this command has been altered by the following extension: containerapp

Name Location ResourceGroup Fqdn

------ ---------- --------------- -----------------------------------------------------

app1 East US default app1.orangetree-90dcdba8.eastus.azurecontainerapps.io

**# We now have 'drift', Azure != local IaC. Show the diff.**

**$** **azx diff aca-app/app1**

Diff "aca-app/app1": < local > azure

### ACA-App(app1) / Containers / #1

< Max Scale: 10

> Max Scale: 9

> Environment variables:

> - abc: def

### ACA-App(app1)

> Bindings:

> - Service: redis1

> Name : redis1

**# Notice:**

**# - it shows it in the context of the human readable output.**

**# - "Max Scale: 10" wasnt set by the user, its a default value**

**#**

**# We can now resolve the diffs via the 'sync' command.**

**# Accept Azure changes(new env var,scale), and reject Azure's old bind**

**$** **azx sync aca-app/app1**

Diff "aca-app/app1": < local > azure

### ACA-App(app1) / Containers / #1

< Max Scale: 10

> Max Scale: 9

a)ccept r)eject azure changes ? a

> Environment variables:

> - abc: def

a)ccept r)eject addition ? a

### ACA-App(app1)

> Bindings:

> - Service: redis1

> Name : redis1

a)ccept r)eject addition ? r

**# Doing a diff still shows the local 'unbind' because we haven't**

**# provisioned back to Azure yet**

**$** **azx diff aca-app/app1**

Diff "aca-app/app1": < local > azure

### ACA-App(app1)

> Bindings:

> - Service: redis1

> Name : redis1

**# So... push it to Azure**

**$** **azx up**

Provision: aca-app/app1

Provision: aca-redis/redis1

**# Now the diff is empty!**

**$** **azx diff aca-app/app1**

**# Demo part 3: Breadcumbs...**

**# - Use the CLI like 'az' but save results in local IaC as we go**

**# - Same basic idea but avoids the manual 'up' action after each step**

**# Update the env var locally AND immediately in Azure**

**$** **azx update aca-app -n app1 --env abc=DEF --up**

Provision: aca-app/app1

**# Prove it's in Azure**

**$** **azx show aca-app -n app1 --from azure**

Name : app1

Environment : /subscriptions/fe108f6a-2bd6-409c-8bfb-8f21dbb7ba0a/resourceGroups/default/providers/Microsoft.App/managedEnvironments/demo

Location : East US

Subscription : fe108f6a-2bd6-409c-8bfb-8f21dbb7ba0a

ResourceGroup: default

Workload Profile Name: Consumption

Ingress: external

Port: 8080

Containers:

- Image : duglin/aca-lister

CPU : 0.5

Memory : 1Gi

Max Scale: 9

Environment variables:

- abc: DEF

**# Turn on auto-provision because I'm getting lazy**

**$** **azx config set defaults.up=true**

**# Now re-bind to redis, happens locally and in Azure**

**$** **azx update aca-app -n app1 --bind redis1**

Provision: aca-app/app1

**# Prove it's locally too**

**$** **azx show aca-app -n app1**

Name : app1

Environment : demo

Location : East US

Subscription : fe108f6a-2bd6-409c-8bfb-8f21dbb7ba0a

ResourceGroup: default

Ingress: external

Containers:

- Image : duglin/aca-lister

Max Scale: 9

Environment variables:

- abc: DEF

Bindings:

- Service: redis1

**# This allows a user, who doesn't immediately care about IaC to**

**# use an AZ-like CLI to interatively construct their app's**

**# config and then save the results in github by saving the**

**# auto-generated IaC files.**

**# They don't need to 'remember' all of the things they set and**

**# recreate the final AZ command or IaC files**

**# Clean up**

**$** **azx down**

Deprovision: aca-app/app1

Deprovision: aca-redis/redis1