



AI-200 Containers 1: ACR, ACR Tasks, App Service

AI-200 skills measured, domain 1 'Develop containerized solutions on Azure' (20-25 %), 'Implement container application hosting', as of 2026 | Build, store, version and manage images in Azure Container Registry; ACR Tasks; containers on App Service with environment variables and secrets

AI-200 Domain 1 ACR ACR Tasks App Service

Recall from AZ-204, then version images

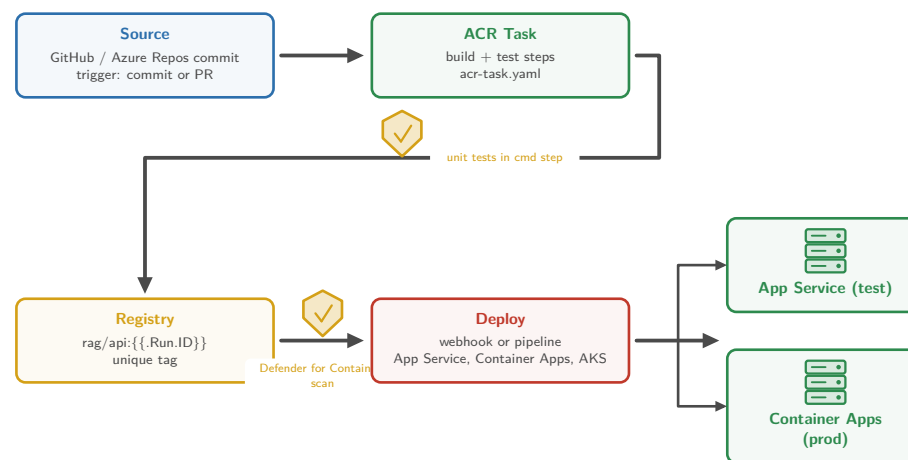
Recall (AZ-204 sheet 03): ACR has three tiers (Basic, Standard, Premium; Premium adds geo-replication, private endpoints, retention policy and more throughput), you authenticate with Entra ID (az acr login, managed identity with **AcrPull** / **AcrPush**) rather than the admin user. AI-200 goes further into **versioning, automated builds and supplying configuration to the container**.



Deploy by unique tag or digest; keep stable tags (latest, 1.4) for humans only

Need	ACR feature	Command / setting
Stop a release tag being overwritten	Lock (write-enabled false)	az acr repository update -n myacr --image rag/api:1.4.2 --write-enabled false
Delete old untagged layers	acr purge in a timer task	acr purge --filter 'rag/api:.*' --ago 7d --untagged
Keep untagged manifests N days (Premium)	Retention policy	az acr config retention update --status enabled --days 7 --type UntaggedManifests
Copy a public image in	Import (no Docker needed)	az acr import -n myacr --source docker.io/library/python:3.12-slim
List tags newest first	Repository API	az acr repository show-tags -n myacr --repository rag/api --orderby time_desc
Pull from AKS	Attach ACR	az aks update -n aks1 -g rg --attach-acr myacr (grants AcrPull to the kubelet identity)
Pull in another region fast	Geo-replication (Premium)	az acr replication create -r myacr -l westeurope

ACR Tasks: build in Azure



Triggers: source commit, base image update (OS patch), timer (cron); runs need no local Docker

az acr task create -r <REGISTRY> -n <TASK> -c <CONTEXT> -f

<FILE> [-t <IMAGE:TAG>] [--git-access-token <PAT>] [--schedule <CRON>]

[--base-image-trigger-enabled true|false] [--commit-trigger-enabled true|false]

command = type exactly literal = type exactly <PLACEHOLDER> = you supply a value

[optional] = may be omitted

REGISTRY --- registry name, not the login server

TASK --- task name

CONTEXT --- Git URL (#branch:folder) or local path

FILE --- Dockerfile or acr-task.yaml (multi-step)

IMAGE:TAG --- use {{.Run.ID}} for a unique tag

CRON --- UTC cron, e.g. '0 3 * * *

acr-task.yaml (multi-step)

1

version: v1.1.0

2

steps:

3

- build: -t \$Registry/rag/api:{{.Run.ID}} -f Dockerfile .

4

- cmd: \$Registry/rag/api:{{.Run.ID}} pytest -q

5

- push:

6

- \$Registry/rag/api:{{.Run.ID}}

7

- cmd: mcr.microsoft.com/acr/acr-cli:0.5 purge --filter 'rag/api:.*' --ago 7d --untagged timeout: 600

8

1

build in Azure; \$Registry = this registry's login server

2

cmd runs a container: tests fail the run before any push

3

push only after tests pass

4

housekeeping step with a timeout in seconds

az acr build -r myacr -t rag/api:{{.Run.ID}} .

Quick task: upload context, build and push in Azure (no local Docker)

az acr run -r myacr -f acr-task.yaml .

Run a multi-step file once

az acr task run -r myacr -n api-ci

Start a defined task manually

az acr task logs -r myacr --run-id cb12

Stream or read a run's log

az acr task list-runs -r myacr -o table

Run history with status and trigger

az acr task identity assign -r myacr -n api-ci

Give the task a managed identity (Key Vault, other registries)

Containers on App Service

Create a Linux web app from ACR with managed-identity pull

```
az appservice plan create -g rg -n plan-rag --is-linux --sku P1v3
az webapp create -g rg -p plan-rag -n rag-api \
  --container-image-name myacr.azurecr.io/rag/api:1.4.2-a1b2c3d \
  --assign-identity [system]
PID=$(az webapp identity show -g rg -n rag-api --query principalId -o tsv)
az role assignment create --assignee $PID --role AcrPull \
  --scope $(az acr show -n myacr --query id -o tsv)
az webapp config set -g rg -n rag-api \
  --generic-configurations '{"acrUseManagedIdentityCreds": true}'
az webapp config appsettings set -g rg -n rag-api --settings \
  WEBSITES_PORT=8000 \
  COSMOS_URL=https://cosmos-rag.documents.azure.com:443/ \
  REDIS_PWD="@Microsoft.KeyVault(VaultName=kv-rag;SecretName=redis-pwd)"
```

Setting or feature	Effect for the container
App settings	injected as environment variables at container start; a change restarts the app
@Microsoft.KeyVault(SecretUri=https://kv-rag.vault.azure.net/secrets/redis-pwd/)	resolved by the app's managed identity (role Key Vault Secrets User); omit the version to pick up rotations (refreshed within 24 h or on restart)
Connection strings	env vars prefixed by type, e.g. CUSTOMCONNSTR_, SQLAZURECONNSTR_
WEBSITES_PORT	port the container listens on when not 80 / 8080
WEBSITES_ENABLE_APP_SERVICE_STORAGE=true	mounts persistent /home storage into the container
Slot setting (sticky)	stays with the slot during swap (recall AZ-204 sheet 01)

Sidecars (Linux)	extra containers, e.g. OpenTelemetry collector, reachable on localhost
Continuous deployment	ACR webhook redeploys on push of the configured tag

Site shows 'Application Error'; log says container didn't respond to HTTP pings

HIGHCOMMON

LIKELY CAUSES

1. App listens on 8000 but WEBSITES_PORT not set

2. App binds to 127.0.0.1 instead of 0.0.0.0

3. Start takes longer than the startup time limit

HOW TO CONFIRM

Read the Docker log stream

az webapp log tail -g rg -n rag-api

FIX

Set WEBSITES_PORT

Bind gunicorn/uvicorn to 0.0.0.0

Raise WEBSITES_CONTAINER_START_TIME_LIMIT

ImagePullFailure / unauthorized from myacr.azurecr.io

HIGHCOMMON

LIKELY CAUSES

1. Identity lacks AcrPull on the registry

2. acrUseManagedIdentityCreds not enabled

3. Registry behind a private endpoint, app without VNet integration

HOW TO CONFIRM

List role assignments of the app identity

az role assignment list --assignee \$PID --all -o table

FIX

Grant AcrPull at registry scope

Enable managed-identity pull

Add VNet integration and private DNS

App setting shows the raw @Microsoft.KeyVault(...) string

MEDIUMOCCASIONAL

LIKELY CAUSES

1. No managed identity or no Key Vault Secrets User role

2. Key Vault firewall blocks the app

HOW TO CONFIRM

Portal: Environment variables, Key Vault reference status

az webapp config appsettings list -g rg -n rag-api

FIX

Assign identity + role

Allow trusted services or use VNet integration

© Worked example: registry storage

PROBLEM A Standard registry (100 GiB included storage) receives 20 CI builds a day; each build adds about 0.25 GiB of new layers. How much storage after 30 days without cleanup, and with a nightly purge of untagged images older than 7 days?

GIVEN

- builds per day $b = 20$
- new layers per build $s = 0.25$ GiB
- horizon = 30 days
- purge keeps the last 7 days
- included storage $I = 100$ GiB (Standard; Basic 10, Premium 500)

1 Growth per day

$g = b \times s = 20 \times 0.25 = 5$ GiB/day

Why: only new layers count, shared layers are stored once

2 No cleanup, 30 days

$5 \times 30 = 150$ GiB

Why: over the 100 GiB included: 50 GiB billed as extra storage

3 With 7-day purge

$5 \times 7 = 35$ GiB (+ tagged releases)

Why: untagged manifests older than 7 days are deleted with their unique layers

ANSWER 150 GiB (50 GiB over) without cleanup; about 35 GiB with the purge task

CHECK ✓ Lock release tags first so the purge never deletes a deployed image

🚩 Recap and self-test

YOU MAY HEAR

"Tag latest is fine for production"

BUSTED

✓ ACTUALLY

Deploy a unique tag or digest

latest moves; rollbacks and audits need an immutable reference.

YOU MAY HEAR

"ACR Tasks needs a build agent"

BUSTED

✓ ACTUALLY

Builds run on ACR compute

az acr build uploads the context; agent pools are optional (Premium) for VNet access.

YOU MAY HEAR

"Key Vault references copy the secret into app settings"

BUSTED

✓ ACTUALLY

They resolve at runtime

The setting stores the reference; the identity reads the secret and the app sees an env var.

Key takeaways

01

Unique tags, locked releases

digest for absolute pinning

02

ACR Tasks automate builds

commit, base image update, timer triggers

03

Clean up untagged manifests

acr purge or retention policy (Premium)

04

Pull with managed identity

AcrPull + acrUseManagedIdentityCreds

05

Config = env vars

app settings, Key Vault references, WEBSITES_PORT

Q1 Command that builds and pushes an image in Azure without local Docker?

A az acr build -r <registry> -t <image:tag> . (a quick task).

Q2 Three ACR Tasks trigger types?

A Source code commit (or PR), base image update, timer (schedule).

Q3 Placeholder for a unique tag per run in a task?

A {{.Run.ID}}.

Q4 File format of a multi-step task and its three step kinds?

A acr-task.yaml with build, push and cmd steps.

Q5 How do you stop a release tag being overwritten or deleted?

A az acr repository update --write-enabled false (and --delete-enabled false).

Q6 Key Vault reference syntax in an app setting?

A @Microsoft.KeyVault(SecretUri=...) or @Microsoft.KeyVault(VaultName=...;SecretName=...).

Q7 Role the web app identity needs to pull from ACR?

A AcrPull on the registry.

Q8 Container listens on port 8000: what must you set?

A App setting WEBSITES_PORT=8000.

Q9 Which tier adds geo-replication and retention policies?

A Premium.