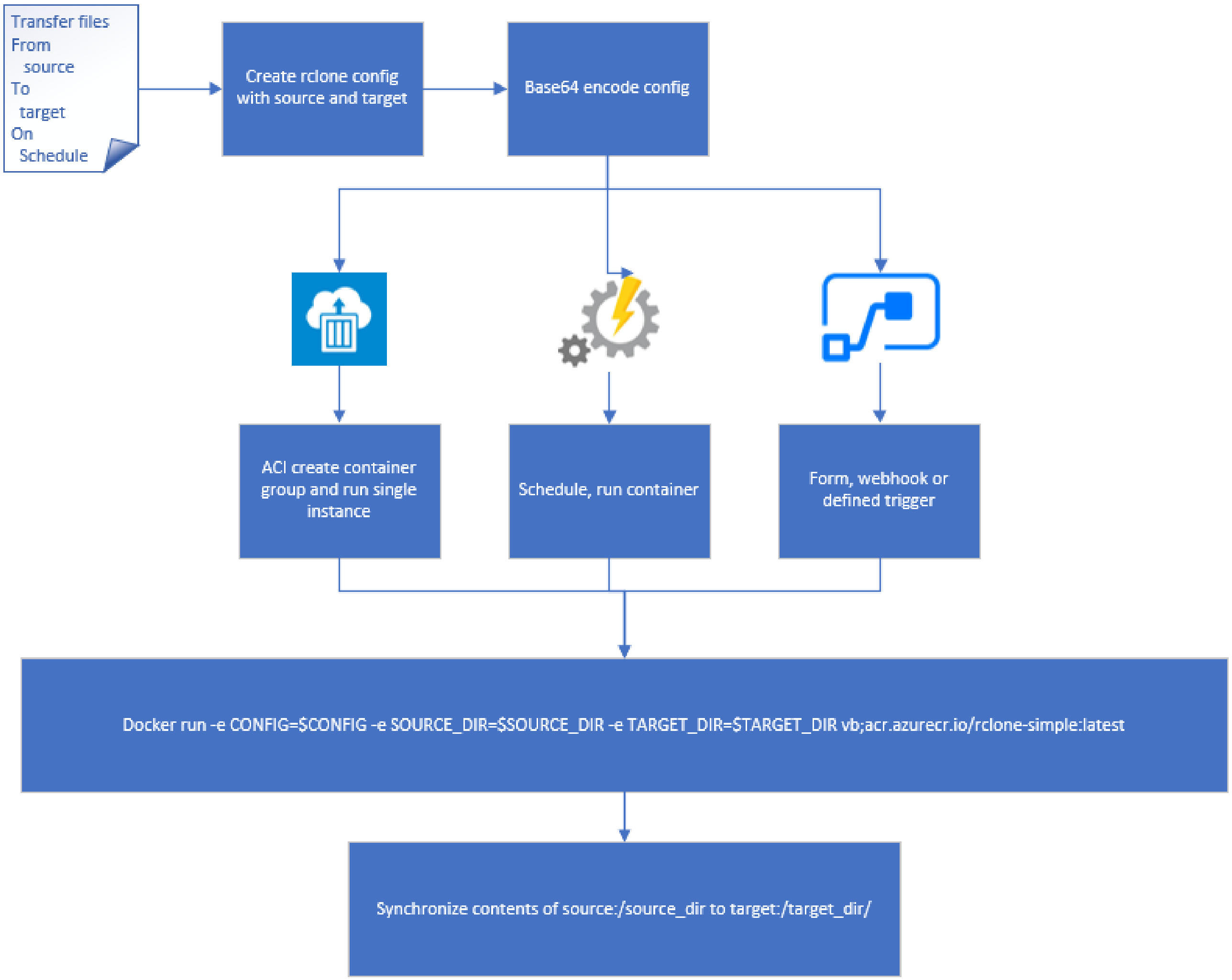
Containerised rclone-simple

Last updated by | Paul Kelleher | 17 Jun 2020 at 07:24 GMT

Introduction

The container listed here can be launched with 3 environment variables and will sync from source to destination


Transfer files
From
source
To
target
On
Schedule
Create rclone config with source and target
Base64 encode config
ACI create container group and run single instance
Schedule, run container
Form, webhook or defined trigger
Synchronize contents of source:/source_dir to target:/target_dir/

```
Docker run -e CONFIG=$CONFIG -e SOURCE_DIR=$SOURCE_DIR -e TARGET_DIR=$TARGET_DIR vb;acr.azurecr.io/rclone-simple:latest
```

Config Paramaters

Environment Variable	Details	Example
CONFIG	The config for rclone, created with "rclone config" creates two remotes. one called source; the other called target. The config will be created under ~/.config/rclone/rclone.conf and shoukld be encoded with base64 (base64 -w0) soi that it can be passed in a as a string. This confif shoudl also be stored in KeyVault as, even though the passwords etc are encrypotred in the config, the entire config itself could be seen as sensitive	
SOURCE_PATH	the location of the files to sync	/user/data/
DEST_PATH	the destination of the files	/pkudata/

Manual Container Launch

The container is stored in the Data Engineering subscription Azure Container registry/repository pknw1.azurecr.io/rclone-simple and can be invoked as follows

```
docker run -it --rm -e CONFIG="" -e SOURCE_PATH="" -e DEST_PATH="" pknw1.azurecr.io/rclone-simple
```

which will run the container

Dockerfile and CMD sqxcript

```
FROM ubuntu:latest

RUN apt-get update -y && apt-get install -y python3 curl wget openssl
WORKDIR /tmp
RUN mkdir -p ~/.config/rclone/
RUN wget -O rclone.deb https://downloads.rclone.org/rclone-current-linux-amd64.deb && dpkg -i ./rclone.deb
RUN ln -s /usr/bin/python3 /usr/bin/python
RUN curl -sL https://aka.ms/InstallAzureCLIDeb | bash


ADD runtime-simple.sh /runtime.sh
ENTRYPOINT ["/runtime.sh"]
CMD ["/usr/sbin/bash", "/runtime.sh"]
```

which runs the runtime.sh script

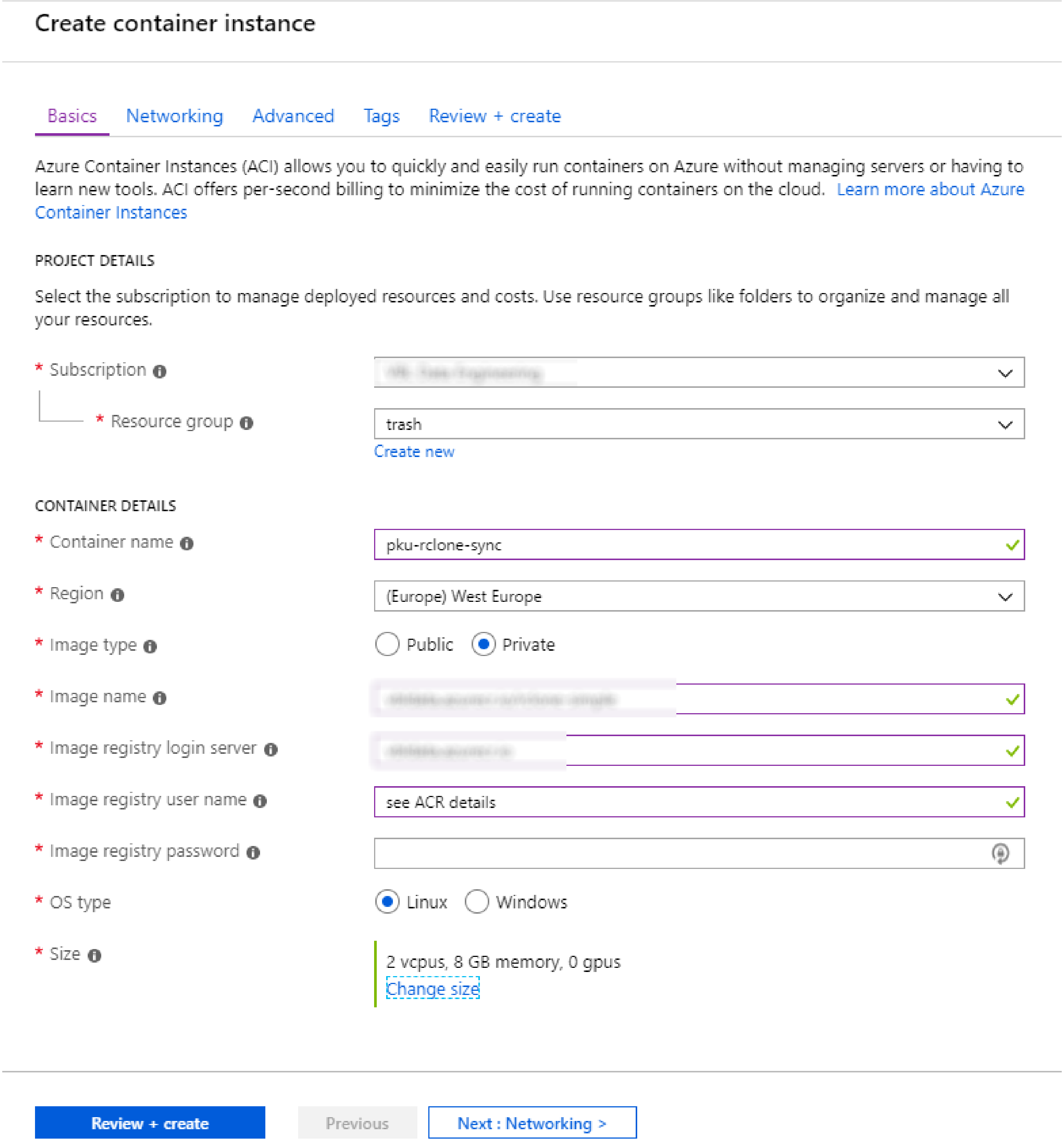
```
#!/bin/bash
TIMESTAMP=$(date +%s)

echo $CONFIG | base64 -d > ~/.config/rclone/rclone.conf

(time rclone sync source:$SOURCE_PATH target:$DEST_PATH --progress && echo  Sync Complete ) || echo There has been an e
```

Running via ACI

The container can be run via ACI and triggered as the application team wants eitehr on schedule or via an automation trigger



Create container instance

Azure Container Instances (ACI) allows you to quickly and easily run containers on Azure without managing servers or having to learn new tools. ACI offers per-second billing to minimize the cost of running containers on the cloud. Learn more about Azure Container Instances

PROJECT DETAILS

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

* Subscription: [redacted]

* Resource group: trash

Create new

CONTAINER DETAILS

* Container name: pku-rclone-sync ✓

* Region: (Europe) West Europe

* Image type: ○ Public ● Private

* Image name: [redacted] ✓

* Image registry login server: [redacted] ✓

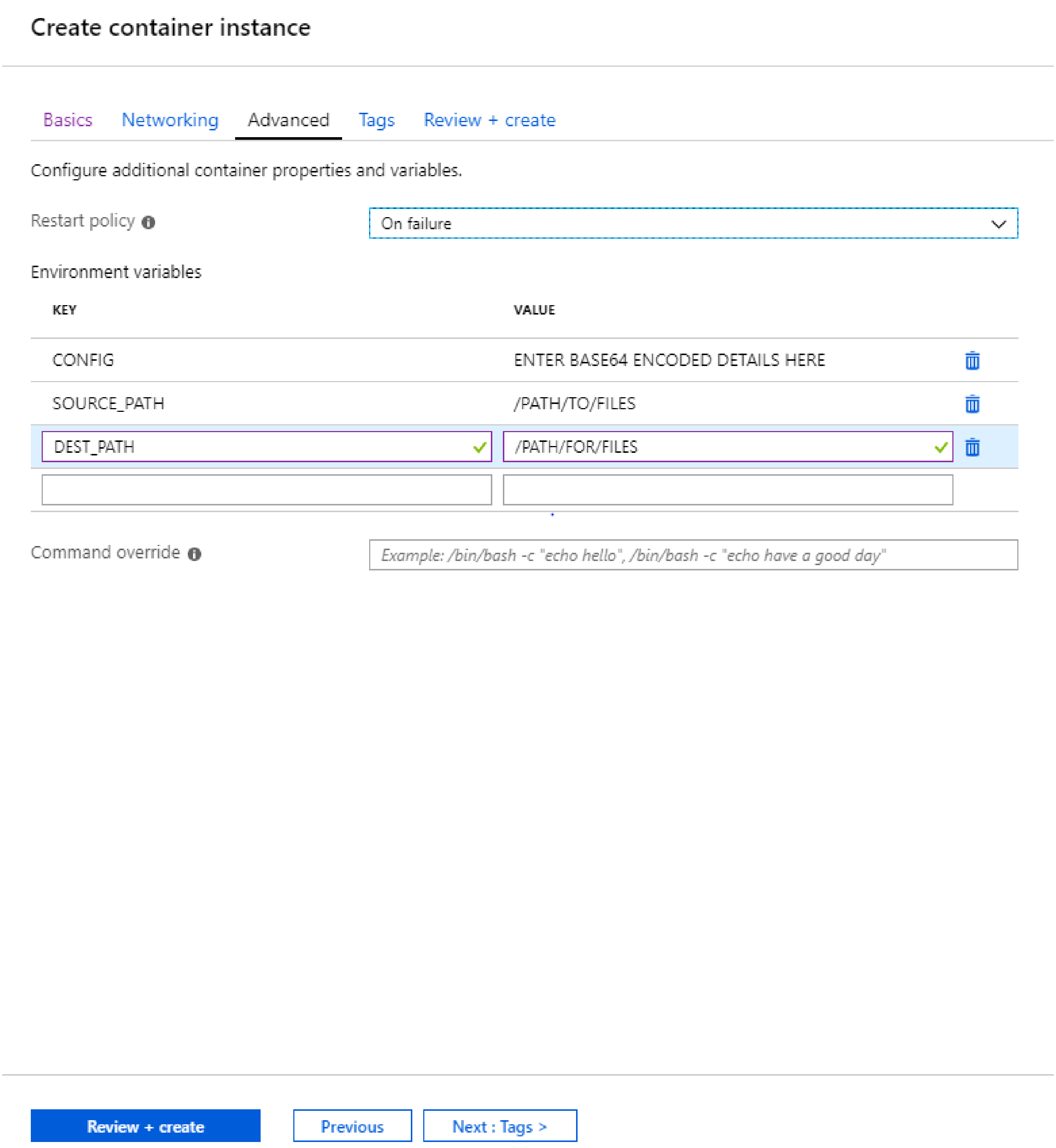
* Image registry user name: see ACR details ✓

* Image registry password:

* OS type: ● Linux ○ Windows

* Size: 2 vcpus, 8 GB memory, 0 gpus

Change size



Create container instance

Configure additional container properties and variables.

Restart policy: On failure

Environment variables

KEY	VALUE
CONFIG	ENTER BASE64 ENCODED DETAILS HERE
SOURCE_PATH	/PATH/TO/FILES
DEST_PATH	/PATH/FOR/FILES

Command override: *Example: /bin/bash -c "echo hello", /bin/bash -c "echo have a good day"*

Contents