



PASS  
MARATHON

# Azure SQL Containers

Running SQL Server containers in Azure

Andrew Pruski

Moderated By: Charith Suriyakula

# Immerse yourself

in the data community

Access deep-dive technical sessions,  
learn best practices, and discover  
new tips and tricks

Gain the technical skills and  
connections to advance  
your data career

# Attend

PASS Summit



PASS Summit is the largest conference for technical professionals who leverage the Microsoft Data Platform.

See everything PASS Summit has to offer at  
[PASSsummit.com](https://PASSsummit.com)

More info:





# Andrew Pruski

SQL Server DBA & Data  
Platform MVP



[dbafromthecold.com](http://dbafromthecold.com)



[@dbafromthecold](https://twitter.com/dbafromthecold)



[dbafromthecold@gmail.com](mailto:dbafromthecold@gmail.com)



PASS  
MARATHON

# Azure SQL Containers

Running SQL Server containers in Azure

Andrew Pruski

Moderated By: Charith Suriyakula

# Agenda

Azure Container Registry

Azure Container Instances

Azure Container Services (ACS & AKS)

# Tools – Docker for Windows

<https://store.docker.com/editions/community/docker-ce-desktop-windows>



## Docker Community Edition for Windows

By **Docker**

The fastest and easiest way to get started with Docker on Windows

Edition

Windows

x86-64

### Get Docker Community Edition for Windows

Docker for Windows is available for free.

Requires Microsoft Windows 10 Professional or Enterprise 64-bit. For previous versions get [Docker Toolbox](#).

By downloading this, you agree to the terms of the [Docker Software End User License Agreement](#)

 **Get Docker**

# Tools – Azure CLI

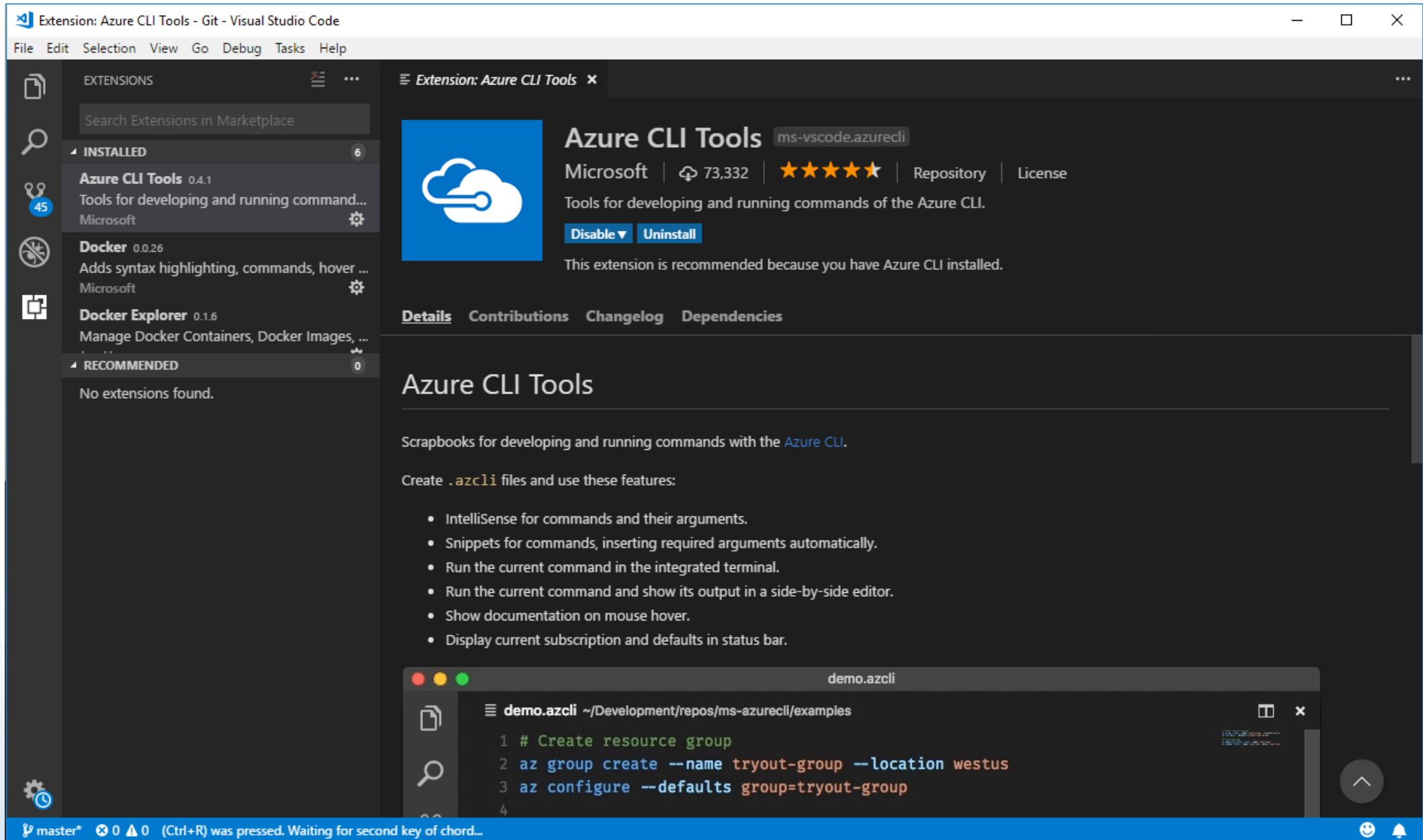
Command line tool for managing Azure resources

Available on Windows, Mac, or Linux (incl. WSL)

<https://docs.microsoft.com/en-us/cli/azure/install-azure-cli?view=azure-cli-latest>

```
andrewpruski@localhost: ~  
andrewpruski@localhost:~$ az --version | grep -e azure-cli -e container  
azure-cli (2.0.30)  
container (0.1.20)  
andrewpruski@localhost:~$ _
```

# Tools – Visual Studio Code



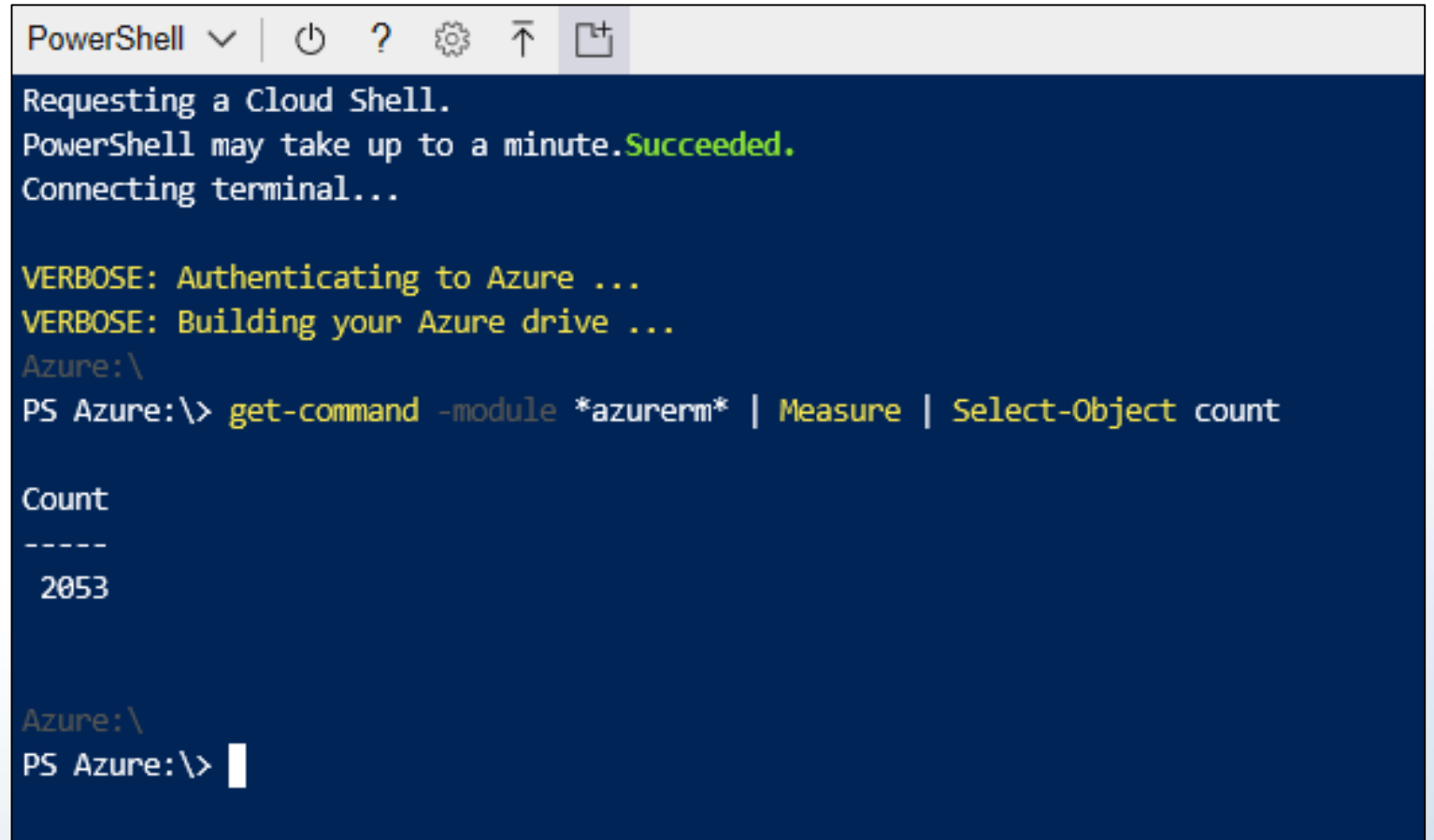


# Tools – Powershell

AzureRM module

Install locally or in CloudShell

Huge amount of commands available



```
PowerShell | [Icons: Refresh, Help, Settings, Up Arrow, Copy]
Requesting a Cloud Shell.
PowerShell may take up to a minute.Succeeded.
Connecting terminal...

VERBOSE: Authenticating to Azure ...
VERBOSE: Building your Azure drive ...
Azure:\
PS Azure:\> get-command -module *azurerm* | Measure | Select-Object count

Count
-----
2053

Azure:\
PS Azure:\> |
```

# Azure Container Registry

# Azure Container Registry

<https://azure.microsoft.com/en-us/services/container-registry/>

## Container Registry

Manage a Docker private registry as a first-class Azure resource

Simplify container development by easily storing and managing container images for Azure deployments in a central registry.

Start free >

Already using Azure? Try Container Registry now >



# Azure Container Registry - Terminology

## Registry

Service that stores container images

## Repository

Groups of container images – Same name, identified by tags

# Azure Container Registry - Features

## Encryption

All images encrypted at rest

## Geo-redundant storage

Replication of images

## Geo-replication

Guards against total regional failure

## ACR Build

Container image build service

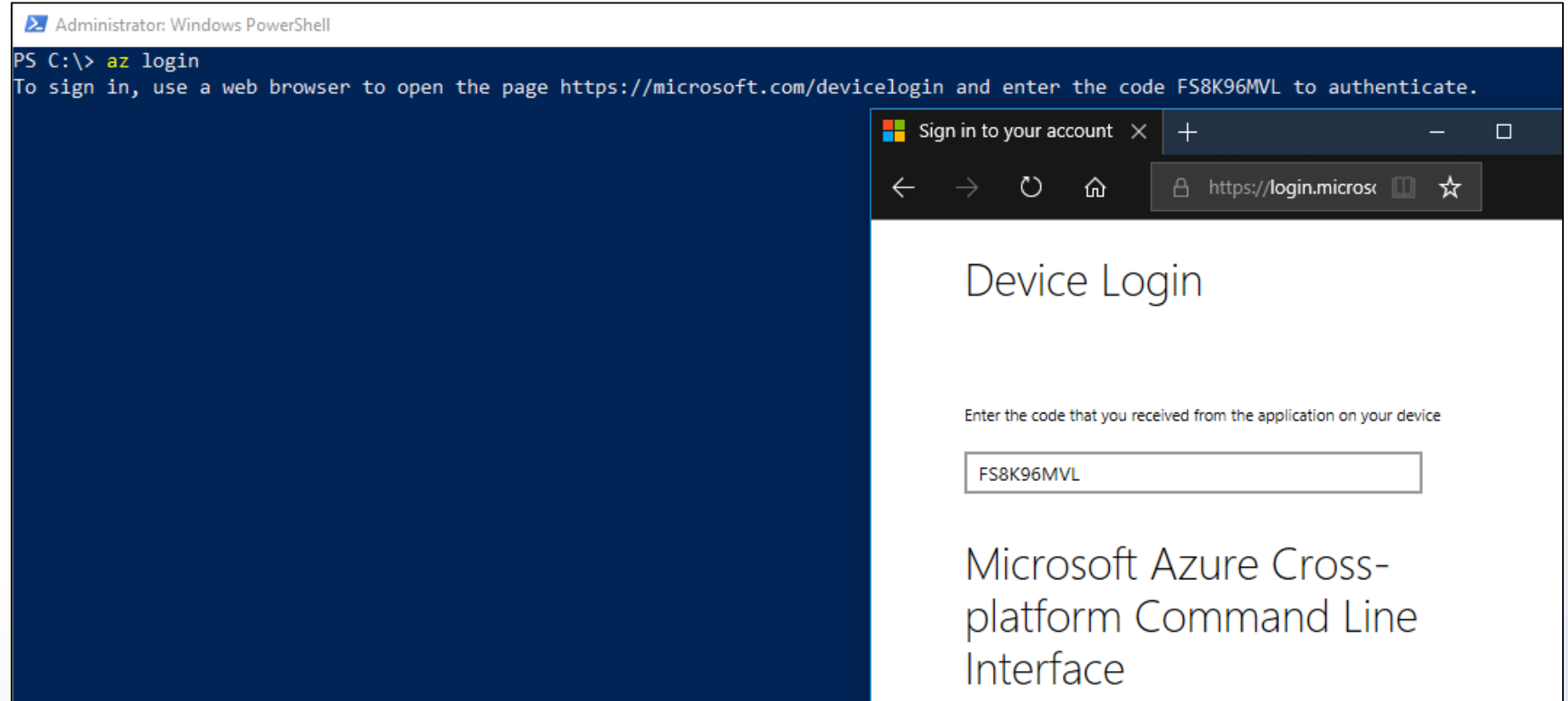
# Azure Container Registry - SKUs

Type	Basic	Standard	Premium
Storage (GB)	10	100	500
ReadOps/min	1000	3000	10000
WriteOps/min	100	500	2000
Download MBps	30	60	100
Upload MBps	10	20	50
Webhooks	2	10	100
Geo-replication	No	No	Yes

<https://docs.microsoft.com/en-us/azure/container-registry/container-registry-skus>

# ACR - Login

az login



# ACR – Create Resource Group

```
az group create --name containers1 --location eastus
```

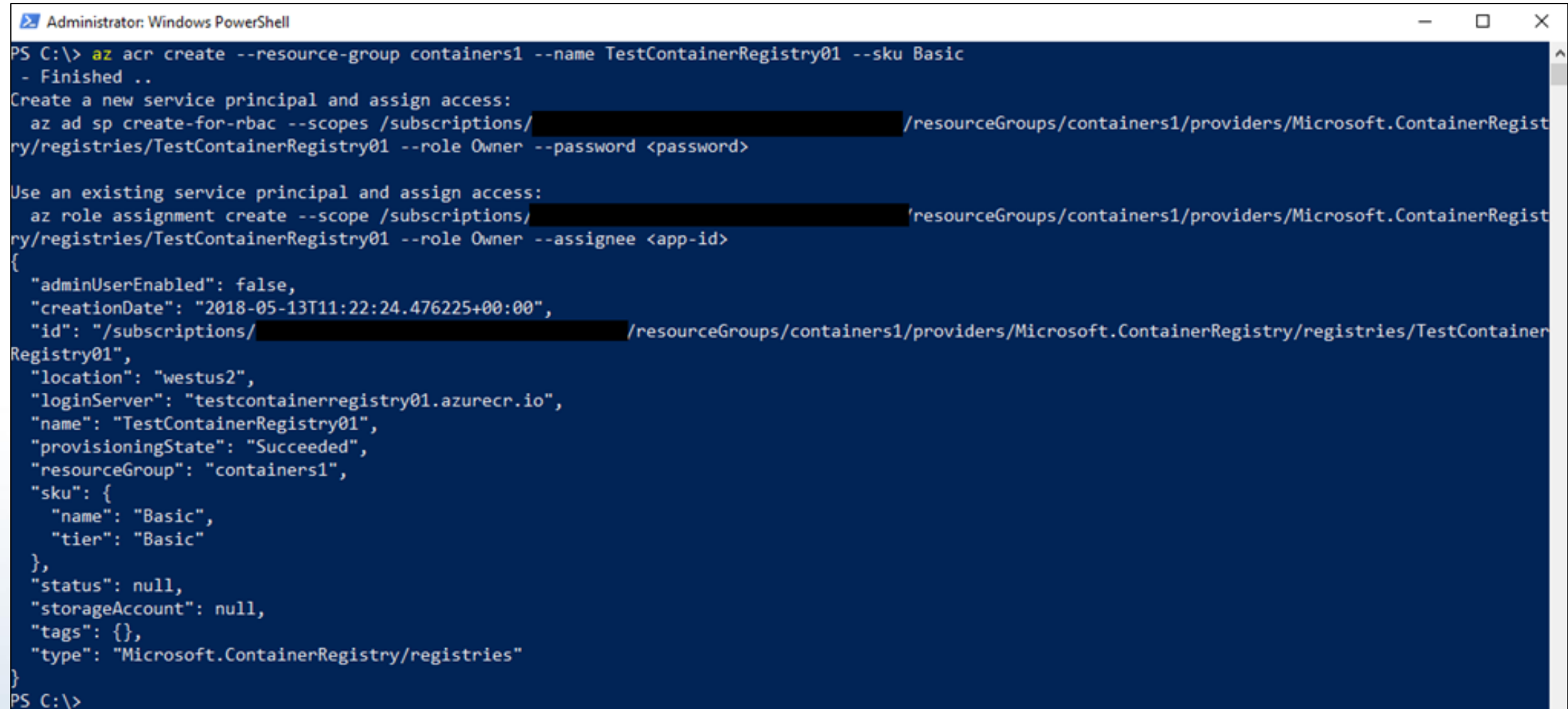
Administrator: Windows PowerShell

```
PS C:\> az group create --name containers1 --location eastus
{
  "id": "/subscriptions/[REDACTED]/resourceGroups/containers1",
  "location": "eastus",
  "managedBy": null,
  "name": "containers1",
  "properties": {
    "provisioningState": "Succeeded"
  },
  "tags": null
}
PS C:\>
```



# ACR – Create Container Registry

```
az acr create --resource-group containers1 --name TestContainerRegistry01 --sku Basic
```



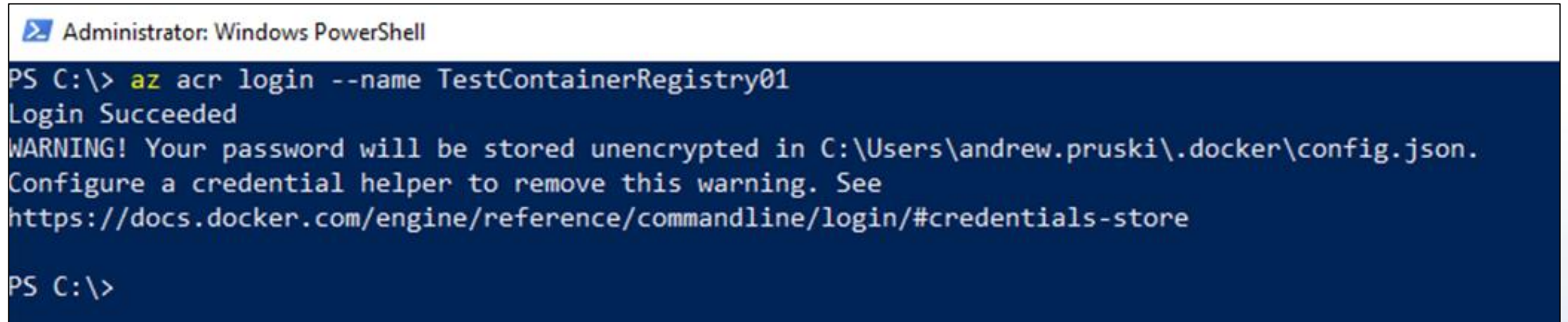
Administrator: Windows PowerShell

```
PS C:\> az acr create --resource-group containers1 --name TestContainerRegistry01 --sku Basic
- Finished ..
Create a new service principal and assign access:
  az ad sp create-for-rbac --scopes /subscriptions/[REDACTED]/resourceGroups/containers1/providers/Microsoft.ContainerRegistry/registries/TestContainerRegistry01 --role Owner --password <password>

Use an existing service principal and assign access:
  az role assignment create --scope /subscriptions/[REDACTED]/resourceGroups/containers1/providers/Microsoft.ContainerRegistry/registries/TestContainerRegistry01 --role Owner --assignee <app-id>
{
  "adminUserEnabled": false,
  "creationDate": "2018-05-13T11:22:24.476225+00:00",
  "id": "/subscriptions/[REDACTED]/resourceGroups/containers1/providers/Microsoft.ContainerRegistry/registries/TestContainerRegistry01",
  "location": "westus2",
  "loginServer": "testcontainerregistry01.azurecr.io",
  "name": "TestContainerRegistry01",
  "provisioningState": "Succeeded",
  "resourceGroup": "containers1",
  "sku": {
    "name": "Basic",
    "tier": "Basic"
  },
  "status": null,
  "storageAccount": null,
  "tags": {},
  "type": "Microsoft.ContainerRegistry/registries"
}
PS C:\>
```

# ACR – Log into Registry

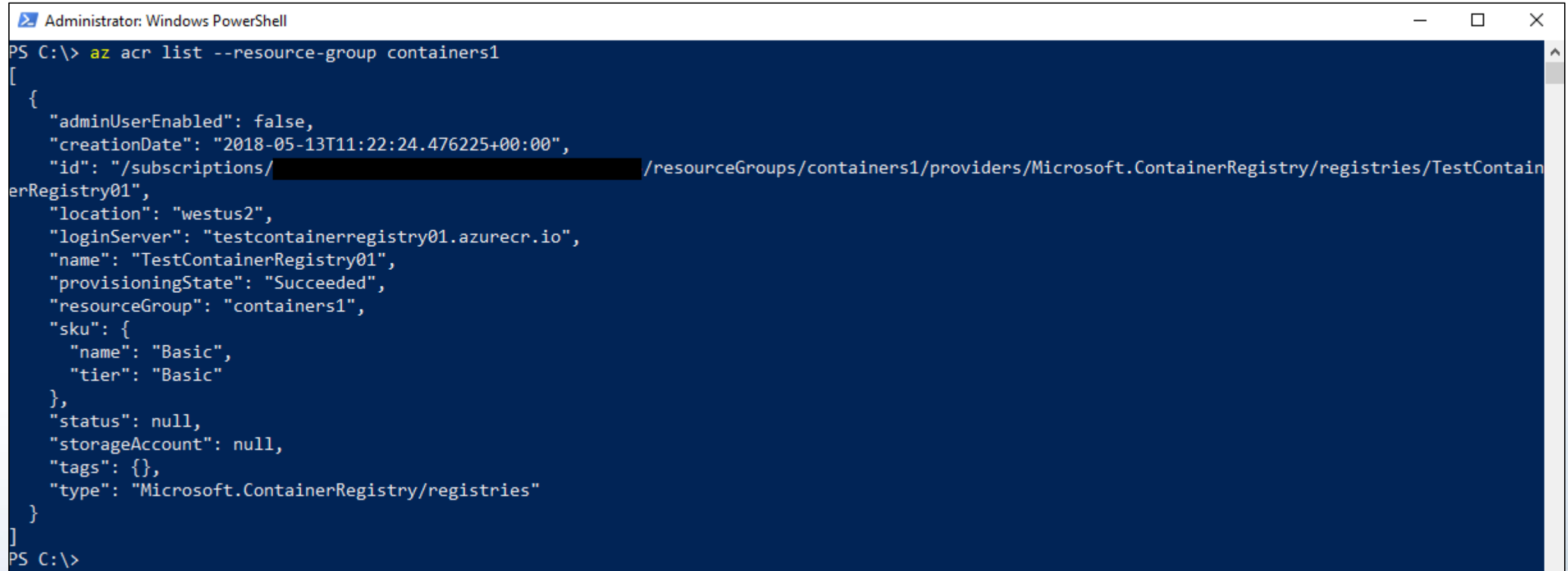
```
az acr login --name TestContainerRegistry01
```



```
Administrator: Windows PowerShell
PS C:\> az acr login --name TestContainerRegistry01
Login Succeeded
WARNING! Your password will be stored unencrypted in C:\Users\andrew.pruski\.docker\config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store
PS C:\>
```

# ACR – Get Registry Details

`az acr list --resource-group containers1`



```
Administrator: Windows PowerShell
PS C:\> az acr list --resource-group containers1
[
  {
    "adminUserEnabled": false,
    "creationDate": "2018-05-13T11:22:24.476225+00:00",
    "id": "/subscriptions/[REDACTED]/resourceGroups/containers1/providers/Microsoft.ContainerRegistry/registries/TestContainerRegistry01",
    "location": "westus2",
    "loginServer": "testcontainerregistry01.azurecr.io",
    "name": "TestContainerRegistry01",
    "provisioningState": "Succeeded",
    "resourceGroup": "containers1",
    "sku": {
      "name": "Basic",
      "tier": "Basic"
    },
    "status": null,
    "storageAccount": null,
    "tags": {},
    "type": "Microsoft.ContainerRegistry/registries"
  }
]
PS C:\>
```

# ACR – Dockerfile

**FROM** microsoft/mssql-server-linux:latest

**CMD** mkdir /var/opt/sqlserver

**COPY** DatabaseA.mdf /var/opt/sqlserver

**COPY** DatabaseA\_log.ldf /var/opt/sqlserver

**ENV** MSSQL\_BACKUP\_DIR="/var/opt/sqlserver"

**ENV** MSSQL\_DATA\_DIR="/var/opt/sqlserver"

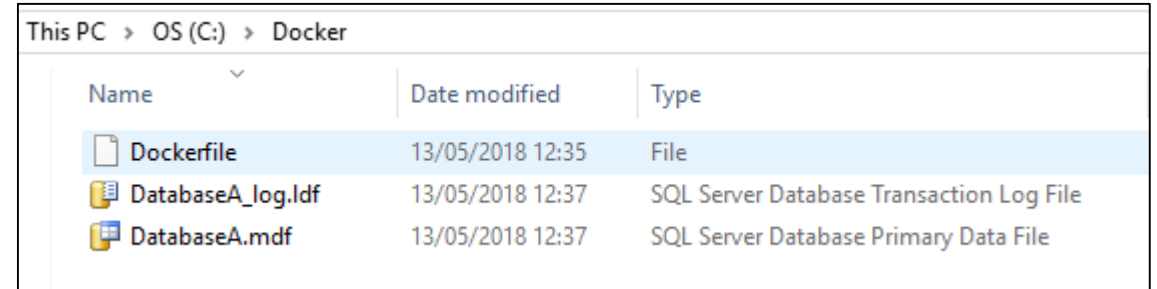
**ENV** MSSQL\_LOG\_DIR="/var/opt/sqlserver"

**HEALTHCHECK** --interval=10s \

**CMD** /opt/mssql/bin/sqlservr & \

/opt/mssql-tools/bin/sqlcmd -S . -U sa -P \$SA\_PASSWORD \

-Q "CREATE DATABASE [DatabaseA] ON (FILENAME = '/var/opt/sqlserver/DatabaseA.mdf'),  
(FILENAME = '/var/opt/sqlserver/DatabaseA\_log.ldf') FOR ATTACH"



This PC > OS (C:) > Docker		
Name	Date modified	Type
Dockerfile	13/05/2018 12:35	File
DatabaseA_log.ldf	13/05/2018 12:37	SQL Server Database Transaction Log File
DatabaseA.mdf	13/05/2018 12:37	SQL Server Database Primary Data File

# ACR – Build image

`docker build -t testimage C:\docker`

```
Administrator: Windows PowerShell
PS C:\> docker build -t testimage C:\docker
Sending build context to Docker daemon 16.78MB
Step 1/8 : FROM microsoft/mssql-server-linux:latest
----> d914e34f3a77
Step 2/8 : RUN mkdir /var/opt/sqlserver
----> Running in 426429c75657
Removing intermediate container 426429c75657
----> e38f4516df30
Step 3/8 : COPY DatabaseA.mdf /var/opt/sqlserver
----> 4a3bce6da149
Step 4/8 : COPY DatabaseA_log.ldf /var/opt/sqlserver
----> 9d75d3d2efce
Step 5/8 : ENV MSSQL_BACKUP_DIR="/var/opt/sqlserver"
----> Running in f2abb95e91f7
Removing intermediate container f2abb95e91f7
----> 13a5c79f0808
Step 6/8 : ENV MSSQL_DATA_DIR="/var/opt/sqlserver"
----> Running in 1cec1ed80a1f
Removing intermediate container 1cec1ed80a1f
----> 31ad98e47754
Step 7/8 : ENV MSSQL_LOG_DIR="/var/opt/sqlserver"
----> Running in 3f40faa1a9d9
Removing intermediate container 3f40faa1a9d9
----> 528d7d166840
Step 8/8 : HEALTHCHECK --interval=10s CMD /opt/mssql/bin/sqlservr & /opt/mssql-tools/bin/sqlcmd -S . -U sa -P $SA_PASSWORD -Q "CREATE DATABASE [DatabaseA] ON (FILENAME = '/var/opt/sqlserver/DatabaseA.mdf'),(FILENAME = '/var/opt/sqlserver/DatabaseA_log.ldf') FOR ATTACH"
----> Running in 57a1da7b0ed7
Removing intermediate container 57a1da7b0ed7
----> 22b5248f1988
Successfully built 22b5248f1988
Successfully tagged testimage:latest
```

# ACR – Tag Custom Image

```
docker tag testimage \
testcontainerregistry01.azurecr.io/devsqlimage:latest
```

```
Administrator: Windows PowerShell
PS C:\> docker tag testimage testcontainerregistry01.azurecr.io/devsqlimage:latest
PS C:\> docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
testimage	latest	2ea5c57bc385	6 minutes ago	1.44GB
testcontainerregistry01.azurecr.io/devsqlimage	latest	2ea5c57bc385	6 minutes ago	1.44GB
microsoft/mssql-server-linux	latest	d914e34f3a77	3 weeks ago	1.43GB

```
PS C:\>
```

# ACR – Push Image

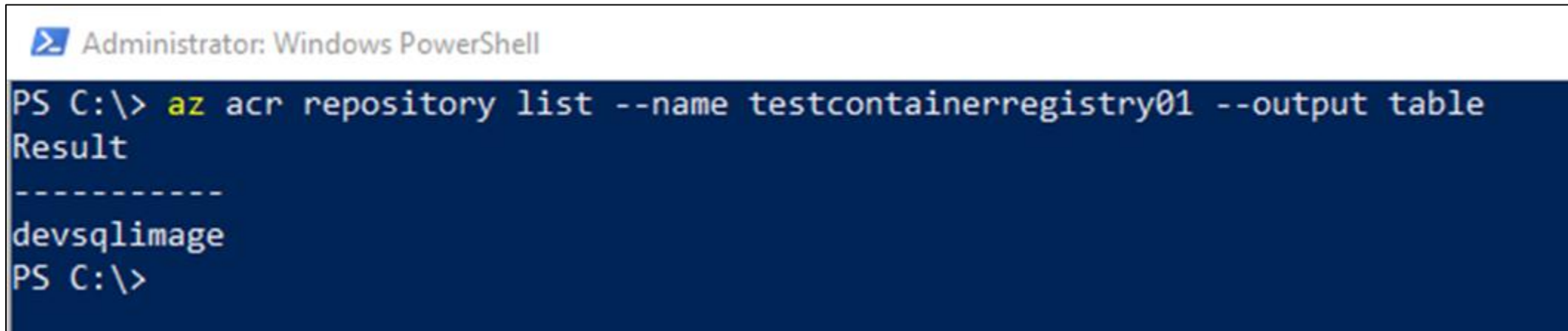
```
docker push testcontainerregistry01.azurecr.io/devsqlimage:latest
```

```
Administrator: Windows PowerShell
PS C:\> docker push testcontainerregistry01.azurecr.io/devsqlimage:latest
The push refers to repository [testcontainerregistry01.azurecr.io/devsqlimage]
0d6e387b3d0b: Pushed
203a1ed50040: Pushed
d3ccabc535f8: Pushed
fb9fa2d3c791: Pushed
0c7dbd94d221: Pushed
45feb6b3c7be: Pushed
912a24c355e6: Pushed
bb83128af95f: Pushed
49907af65b0a: Pushed
4589f96366e6: Pushed
b97229212d30: Pushed
cd181336f142: Pushed
0f5ff0cf6a1c: Pushed
latest: digest: sha256:ad426ffbe36af0d61dcb45732e69f01f27cf0343a18a5c092e654c012fcc298d size: 3039
PS C:\>
```



# ACR – List Images

```
az acr repository list --name testcontainerregistry01 --output table
```



```
Administrator: Windows PowerShell
PS C:\> az acr repository list --name testcontainerregistry01 --output table
Result
-----
devsqlimage
PS C:\>
```



# ACR – Show Tags

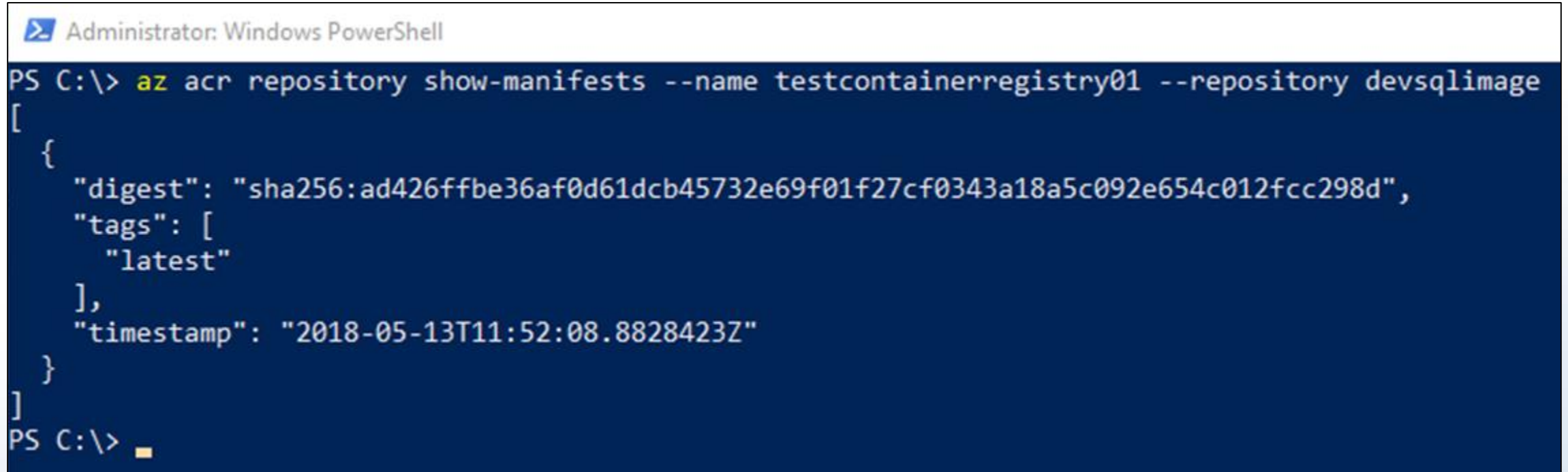
```
az acr repository show-tags \
  --name testcontainerregistry01 \
  --repository devsqlimage
```

Administrator: Windows PowerShell

```
PS C:\> az acr repository show-tags --name testcontainerregistry01 --repository devsqlimage
[
  "latest"
]
PS C:\> █
```

# ACR – Show Manifest

```
az acr repository show-manifests`  
  --name testcontainerregistry01`  
  --repository devsqlimage`
```



The screenshot shows a Windows PowerShell terminal window titled "Administrator: Windows PowerShell". The command entered is `az acr repository show-manifests --name testcontainerregistry01 --repository devsqlimage`. The output is a JSON array containing one manifest object.

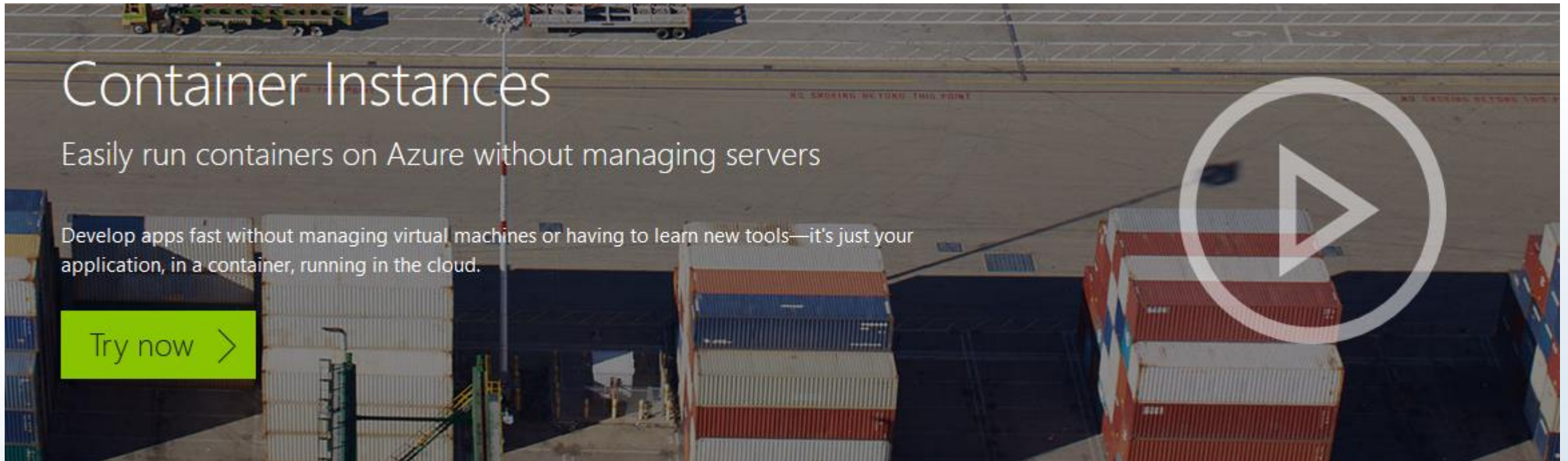
```
PS C:\> az acr repository show-manifests --name testcontainerregistry01 --repository devsqlimage  
[  
  {  
    "digest": "sha256:ad426ffbe36af0d61dcb45732e69f01f27cf0343a18a5c092e654c012fcc298d",  
    "tags": [  
      "latest"  
    ],  
    "timestamp": "2018-05-13T11:52:08.8828423Z"  
  }  
]  
PS C:\> █
```

# Demo

# Azure Container Instances

# Azure Container Instances

<https://azure.microsoft.com/en-us/services/container-instances/>



# Azure Container Instances

Running containers without servers

No need to manage VMs

Quick deployment

Deployed via the CLI, powershell, or Azure Portal

Billed by the second

# ACI - Options

Linux & Windows containers

Containers exposed directly to the internet

IP Address and FQDN

Hypervisor level isolation

Persistent storage

Azure files shares

# ACI – Container Groups

Similar in concept to K8s pods

Multiple containers running on the same host

Share IP address, containers exposed on ports

Supports mounting external volumes



# ACI – Get ACR Credentials

```
# enable admin
```

```
az acr update -n TestContainerRegistry01 --admin-enabled true
```

```
# get credentials
```

```
az acr credential show -n TestContainerRegistry01
```

# ACI – Create Keyvault

```
az keyvault create --resource-group containers1 --name aptestkeyvault01
```

```
az keyvault secret set `
  --vault-name aptestkeyvault01 `
  --name testcontainerregistry01-pull-pwd `
  --value $(az ad sp create-for-rbac `
    --name testcontainerregistry01-pull `
    --scopes $(az acr show --name testcontainerregistry01 --query id --output tsv)`
    --role reader --query password --output tsv)
```

```
az keyvault secret set `
  --vault-name aptestkeyvault01 `
  --name testcontainerregistry01-pull-usr `
  --value $(az ad sp show --id http://testcontainerregistry01-pull `
    --query appId --output tsv)
```

# ACI – Create Container

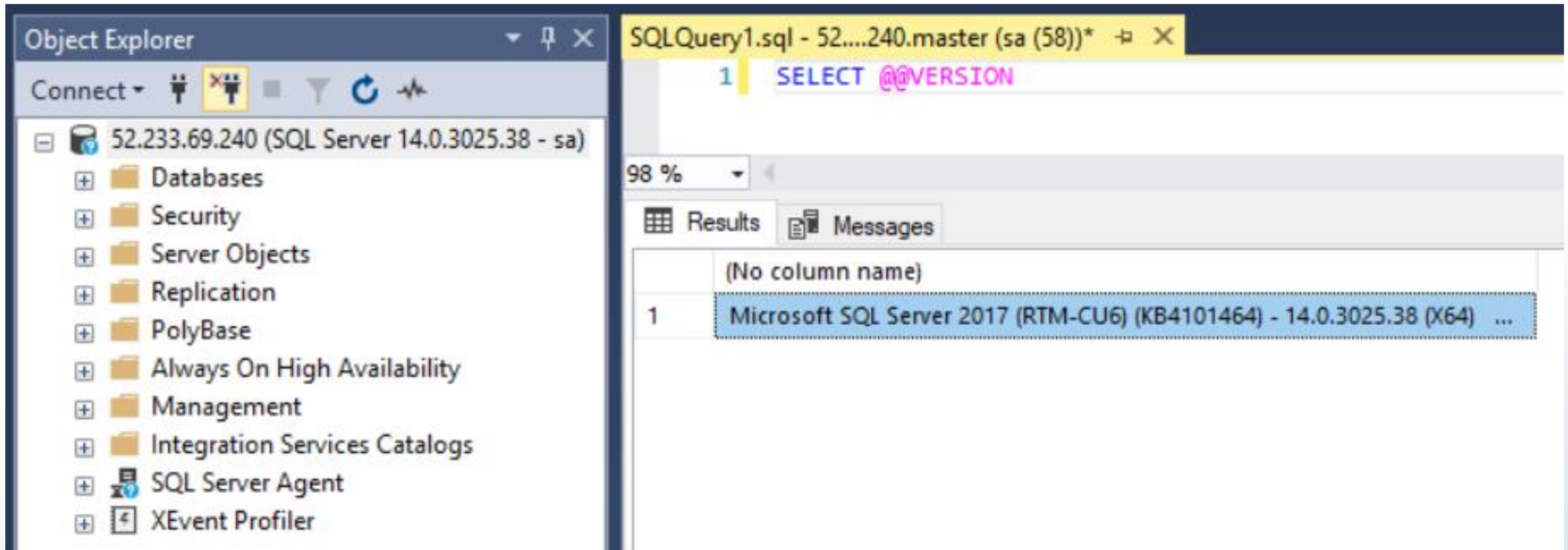
```
az container create `
  --resource-group containers1 `
  --image testcontainerregistry01.azurecr.io/devsqlimage:latest `
  --registry-username $(az keyvault secret show `
    --vault-name aptestkeyvault01 `
    --name testcontainerregistry01-pull-usr `
    --query value --output tsv) `
  --registry-password $(az keyvault secret show `
    --vault-name aptestkeyvault01 `
    --name testcontainerregistry01-pull-pwd `
    --query value --output tsv) `
  --name testcontainer1 `
  --cpu 2 --memory 4 `
  --environment-variables ACCEPT_EULA=Y SA_PASSWORD=Testing1122 `
  --ip-address public `
  --ports 1433
```

# ACI – Get Container Status

```
az container show`  
  --name testcontainer1  
  --resource-group containers1
```





```
Administrator: Windows PowerShell  
  
{  
  "password": null,  
  "server": "testcontainerregistry01.azurecr.io",  
  "username": "testcontainerregistry01"  
},  
"instanceView": {  
  "events": [],  
  "state": "Running"  
},  
"ipAddress": {  
  "ip": "52.233.69.240",  
  "ports": [  
    {  
      "port": 1433,  
      "protocol": "TCP"  
    }  
  ]  
},  
"location": "westus2",  
"name": "testcontainer1",  
"osType": "linux",  
"provisioningState": "Succeeded",  
"resourceGroup": "containers1",  
"restartPolicy": "Always",  
"tags": null,  
"type": "Microsoft.ContainerInstance/containerGroups",  
"volumes": null  
}  
PS C:\>
```

# ACI – Connect to SQL Server



The screenshot displays the SQL Server Enterprise Manager interface. On the left, the 'Object Explorer' pane shows a tree view of the server instance '52.233.69.240 (SQL Server 14.0.3025.38 - sa)'. The tree includes folders for Databases, Security, Server Objects, Replication, PolyBase, Always On High Availability, Management, Integration Services Catalogs, SQL Server Agent, and XEvent Profiler. The main window shows a query window titled 'SQLQuery1.sql - 52....240.master (sa (58))\*'. The query text is 'SELECT @@VERSION'. The 'Results' tab is active, showing a single row of results with the column name '(No column name)' and the value 'Microsoft SQL Server 2017 (RTM-CU6) (KB4101464) - 14.0.3025.38 (X64) ...'. The progress bar at the top of the results pane indicates 98% completion.

Object Explorer

Connect ▾    

52.233.69.240 (SQL Server 14.0.3025.38 - sa)

- + Databases
- + Security
- + Server Objects
- + Replication
- + PolyBase
- + Always On High Availability
- + Management
- + Integration Services Catalogs
- + SQL Server Agent
- + XEvent Profiler

SQLQuery1.sql - 52....240.master (sa (58))\*

1 SELECT @@VERSION

98 %

Results Messages

	(No column name)
1	Microsoft SQL Server 2017 (RTM-CU6) (KB4101464) - 14.0.3025.38 (X64) ...

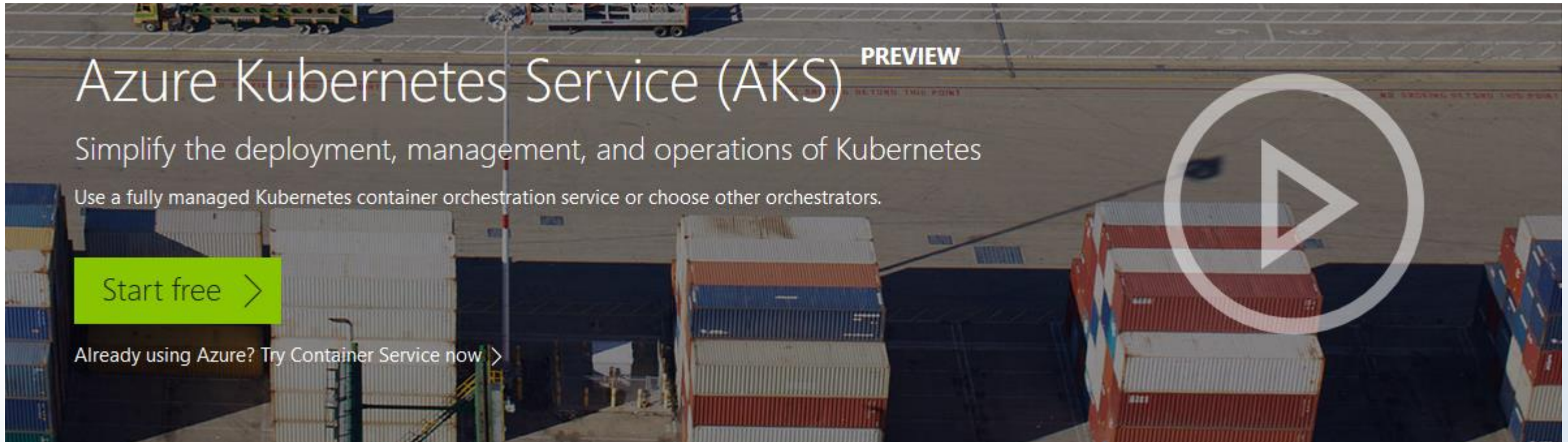
# Demo

# Azure Container Services



# Azure Container Services

<https://azure.microsoft.com/en-us/services/container-service/>

A promotional banner for Azure Kubernetes Service (AKS). The background is an aerial view of a port with many colorful shipping containers (red, blue, white) stacked in rows. In the upper left, a yellow truck is visible. The text is overlaid on the left side. A large, semi-transparent play button icon is on the right side.

**Azure Kubernetes Service (AKS)** PREVIEW

Simplify the deployment, management, and operations of Kubernetes

Use a fully managed Kubernetes container orchestration service or choose other orchestrators.

[Start free >](#)

[Already using Azure? Try Container Service now >](#)



# Azure Container Services

Two flavours

- Azure Container Services (ACS)

- Azure Container Services (AKS)

ACS provides container hosting using DC/OS, Swarm, or K8s

AKS is specifically built to implement Kubernetes

# Kubernetes

Open Source system for managing containers

Deployed as a cluster containing a master and multiple nodes

Pods hold containers running on the nodes

Services define/allow access to sets of pods

Deployments created and managed via Kubectl



# Azure Container Services (AKS)

Simplifies deployment of Kubernetes clusters in Azure

Cluster can be spun up with one line of code

Applications deployed to cluster via yaml files

Managed by Azure-CLI/powershell and kubectl

# AKS – Create Cluster

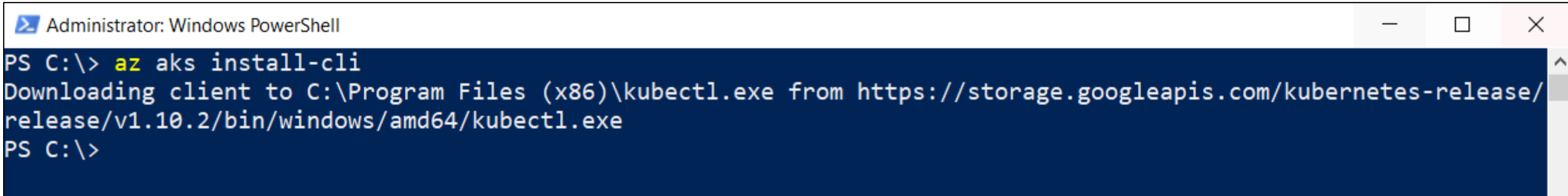
```
az aks create `
  --resource-group containers1 `
  --name mySQLK8sCluster1 `
  --node-count 2 `
  --generate-ssh-keys
```

Administrator: Windows PowerShell

```
PS C:\> az aks create --resource-group containers1 --name mySQLK8sCluster1 --node-count 2 --generate-ssh-keys
- Running ...
```

# AKS – Install kubectl

```
az aks install-cli
```



```
Administrator: Windows PowerShell
PS C:\> az aks install-cli
Downloading client to C:\Program Files (x86)\kubectl.exe from https://storage.googleapis.com/kubernetes-release/release/v1.10.2/bin/windows/amd64/kubectl.exe
PS C:\>
```

# AKS – Get Cluster Credentials

```
az aks get-credentials `
  --resource-group containers1 `
  --name mySQLK8sCluster1
```

```
Administrator: Windows PowerShell
PS C:\> az aks get-credentials --resource-group containers1 --name mySQLK8sCluster1
Merged "mySQLK8sCluster1" as current context in C:\Users\andrew.pruski\.kube\config
PS C:\>
```

# AKS – View Cluster Nodes

`kubectl get nodes`

Administrator: Windows PowerShell

```
PS C:\> kubectl get nodes
```

NAME	STATUS	ROLES	AGE	VERSION
aks-nodepool1-40591065-0	Ready	agent	12m	v1.7.7
aks-nodepool1-40591065-1	Ready	agent	13m	v1.7.7

```
PS C:\>
```

# AKS – Get Cluster & ACR Details

Get AKS Service Principal ID

```
az aks show `
    --resource-group containers1 `
    --name mySQLK8sCluster1 `
    --query "servicePrincipalProfile.clientId" --output tsv
```

Get ACR Resource ID

```
az acr show `
    --name TestContainerRegistry02 `
    --resource-group containers1 `
    --query "id" `
    --output tsv
```



# AKS – Create Role to Deploy

Create role to grant access

```
az role assignment create `
  --assignee <<CLIENTID>> `
  --role Reader `
  --scope <<ACRID>>
```

# AKS – yaml file

```
apiVersion: apps/v1beta1
kind: Deployment
metadata:
  name: sqlserver
  labels:
    app: sqlserver
spec:
  replicas: 1
  template:
    metadata:
      labels:
        name: sqlserver
```

# AKS – yaml file

spec:

containers:

- name: sqlserver1

image: testcontainerregistry01.azurecr.io/devsqlimage:latest

ports:

- containerPort: 1433

env:

- name: SA\_PASSWORD

value: "Testing1122"

- name: ACCEPT\_EULA

value: "Y"

# AKS – yaml file

```
apiVersion: v1
kind: Service
metadata:
  name: sqlserver-service
spec:
  ports:
    - name: sqlserver
      port: 1433
      targetPort: 1433
  selector:
    name: sqlserver
  type: LoadBalancer
```

# AKS – Deploy to Cluster

```
kubectl create -f sqlserver.yml
```

Administrator: Windows PowerShell

```
PS C:\> kubectl apply -f sqlserver.yml  
deployment "sqlserver" created  
service "sqlserver-service" created  
PS C:\> _
```

# AKS – View Deployment Information

kubectl get deployments

kubectl get pods

Kubectl get service

```
Administrator: Windows PowerShell
PS C:\> kubectl get deployment
NAME          DESIRED  CURRENT  UP-TO-DATE  AVAILABLE  AGE
sqlserver     1        1        1           1          9m
PS C:\> kubectl get pods
NAME                                READY  STATUS   RESTARTS  AGE
sqlserver-1093143461-712gs         1/1    Running  0         9m
PS C:\> kubectl get service
NAME              TYPE          CLUSTER-IP    EXTERNAL-IP    PORT(S)          AGE
kubernetes        ClusterIP     10.0.0.1      <none>         443/TCP          46m
sqlserver-service  LoadBalancer 10.0.108.239  168.61.52.8    1433:31462/TCP   9m
PS C:\>
```

# AKS – Connect to SQL Server

The screenshot displays the SQL Server Enterprise Manager interface. On the left, the 'Object Explorer' pane shows a connection to '168.61.52.8 (SQL Server 14.0.3025.38 - sa)'. The tree view includes folders for Databases, Security, Server Objects, Replication, PolyBase, Always On High Availability, Management, Integration Services Catalogs, SQL Server Agent (Agent XPs disabled), and XEvent Profiler. The 'Databases' folder is expanded, showing 'System Databases', 'Database Snapshots', and 'DatabaseA'. The main pane shows a query window titled 'SQLQuery1.sql - 168...2.8.master (sa (53))\*'. The query text is 'SELECT @@VERSION'. The 'Results' tab is active, showing a single row with the value 'Microsoft SQL Server 2017 (RTM-CU6) (KB4101464) - 14.0.3025.38 (X64) ...'. The progress bar at the top indicates 98% completion.

Object Explorer

Connect

168.61.52.8 (SQL Server 14.0.3025.38 - sa)

- Databases
  - System Databases
  - Database Snapshots
  - DatabaseA
- Security
- Server Objects
- Replication
- PolyBase
- Always On High Availability
- Management
- Integration Services Catalogs
- SQL Server Agent (Agent XPs disabled)
- XEvent Profiler

SQLQuery1.sql - 168...2.8.master (sa (53))\*

```
1 SELECT @@VERSION
```


98 %

Results Messages

	(No column name)
1	Microsoft SQL Server 2017 (RTM-CU6) (KB4101464) - 14.0.3025.38 (X64) ...

# AKS – Connect to K8s Dashboard

```
az aks browse --resource-group containers1 --name mySQLK8sCluster1
```

 **kubernetes**

Search

+ CREATE

Overview

Cluster

Namespaces

Nodes

Persistent Volumes

Roles

Storage Classes

Namespace

default

Overview

Workloads

Daemon Sets

Deployments

Jobs

Pods

Deployments

Name	Labels	Pods	Age	Images
✓ <a href="#">sqlserver</a>	app: sqlserver	1 / 1	14 minutes	testcontainerregistry01.azurecr.i...

Pods

Name	Node	Status	Restarts	Age
✓ <a href="#">sqlserver-1093143461-712gs</a>	aks-nodepool1-40591065-0	Running	0	14 minutes

Replica Sets

Name	Labels	Pods	Age	Images
✓ <a href="#">sqlserver-1093143461</a>	name: sqlserver pod-template-hash: 10931434...	1 / 1	14 minutes	testcontainerregistry01.azurecr.i...



# Demo

# Resources

<https://azure.microsoft.com/en-us/services/container-registry/>

<https://azure.microsoft.com/en-us/services/container-instances/>

<https://azure.microsoft.com/en-us/services/container-service/>

<https://dbafromthecold.com/2017/03/15/summary-of-my-container-series/>

<https://github.com/dbafromthecold/ContainersInTheCloud>



# Thank you for attending

Learn more from Andrew Pruski



@dbafromthecold



dbafromthecold@gmail.com



@sqlpass #sqlpass



@PASScommunity



\*PASS

MARATHON