Specifying Deployment Requirements in Microsoft Azure

CHOOSING A DEPLOYMENT METHODOLOGY

Chris B. Behrens
SOFTWARE ARCHITECT

www.chrisbehrens.rocks

Course Overview

Talking about HOW to choose a deployment methodology

Look at what ALL your options are

Walk through them one by one

How This Course Is Organized

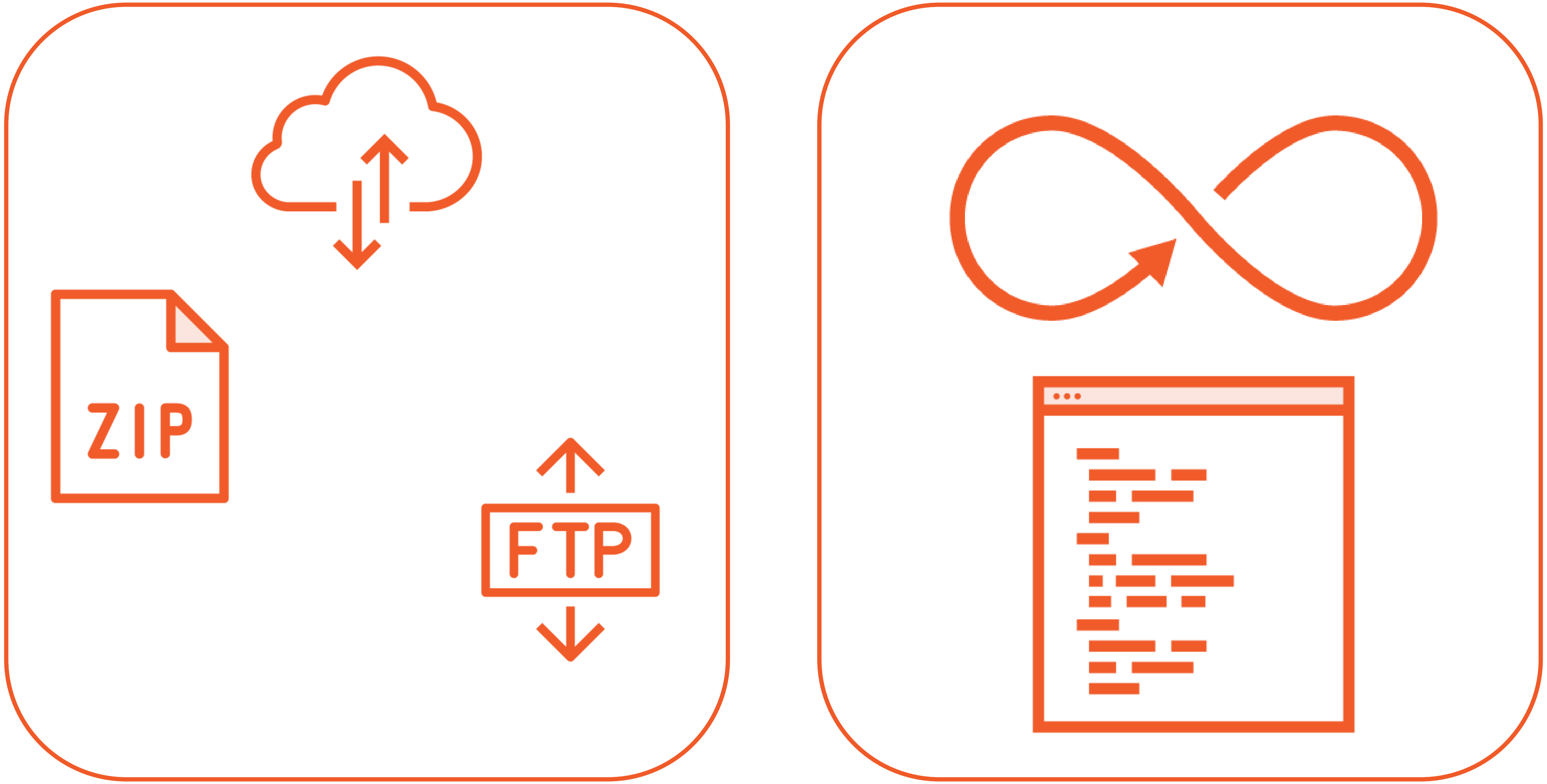
First course in a three-part series

First module implements full, simple deployments

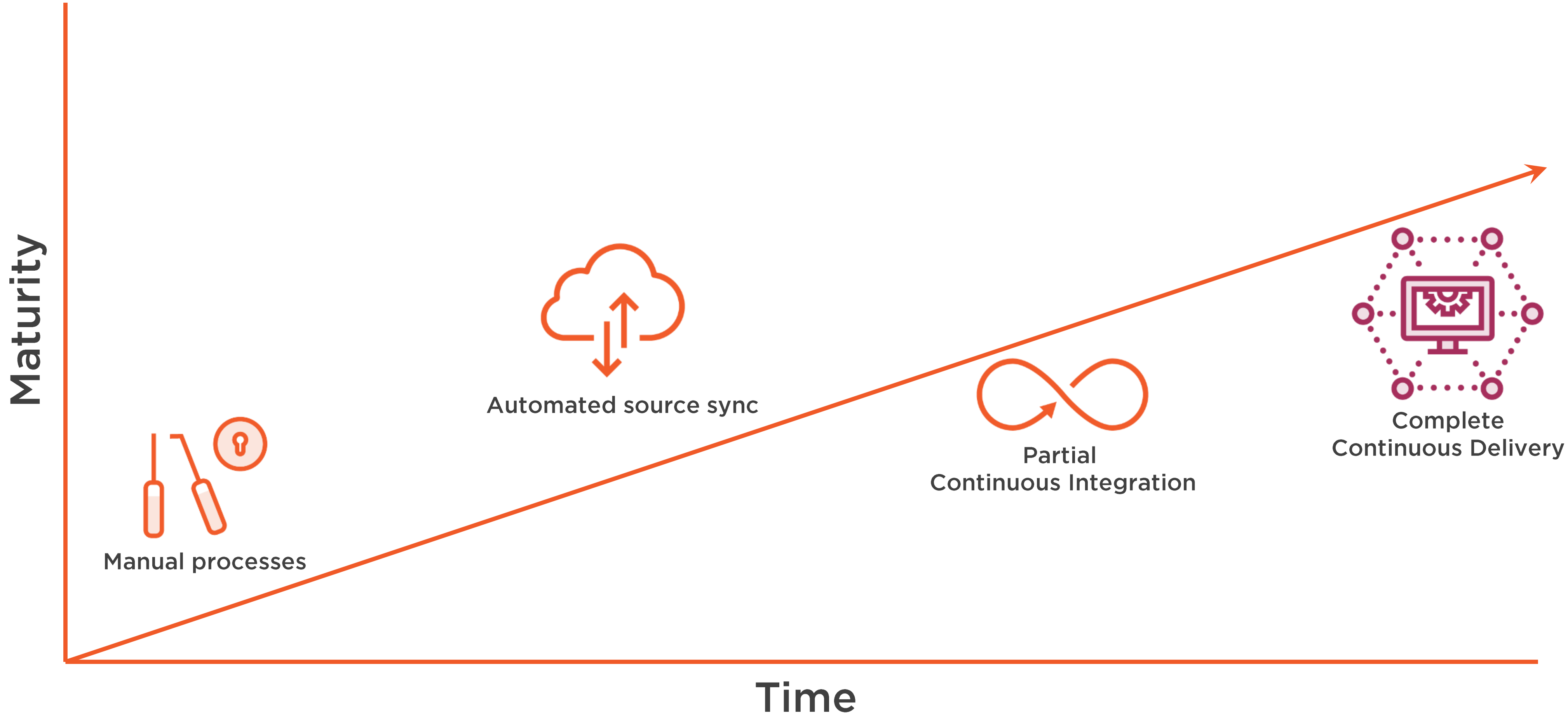
Bottom rung of a big ladder

Leads to state of the art deployments

Choosing a Deployment Methodology



Methodology Reflects Maturity



So, Where Am I?

- GitHub
- BitBucket
- DropBox
- OneDrive

Stress Is a Signal of the Need for Growth

Many methods are simple and quick to implement

But once you feel pain, don’t bother to automate them

Instead, invest that effort in implementing Continuous Integration

Fast-forward for Git People

ARM templates

Database deployment

Infrastructure as Code

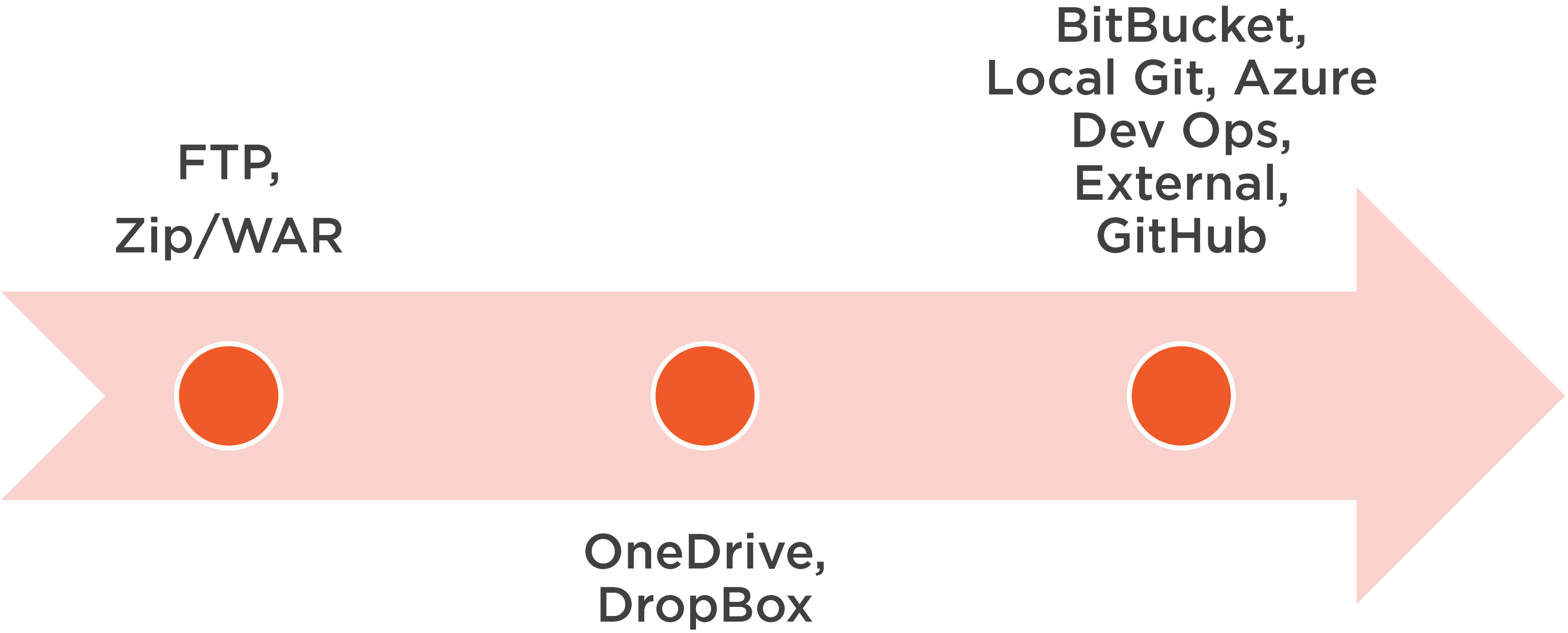
Following Courses

Azure Deployment - Configuring Your Deployment Environment

Azure Deployment - Deploying Web Applications and Services

A Map of the Deployment Options

A Map of the Deployment Options



FTP,

Zip/WAR

OneDrive,
DropBox

BitBucket,
Local Git, Azure
Dev Ops,
External,
GitHub

File Transfer-oriented Methods

1. FTP
2. Zip / WAR files

- Set up host and credentials in your client
- Transfer the files up and down
- Dead simple

Demo

Configure FTP deployment in the portal

Make a small change to our source code

Configure our FTP client to connect to the Azure FTP Endpoint

Deploy and verify that change

Zip and WAR Deployment

Zip Files

- A compressed archive containing your application files
- Deploys to https://<app_name>.scm.azurewebsites.net/ZipDeploy
- Via Portal
- Via command line / script

WAR Files

- Web Application Resource / Web Application Archive
- Deploys to https://<app_name>.scm.azurewebsites.net/api/wardeploy
- Via script POST
- Via command line / script

WAR – What Is It Good For?

Application files

META file which specifies how application is organized

Only for Java applications

How to Deploy Archive Files

Navigate to ZipDeploy folder

- https://<app_name>.scm.azurewebsites.net/ZipDeploy

Drop your archive file there

Demo

Review the ZipDeploy interface in the portal

Create a zip deployment package for our application

Deploy and verify the change

Scripting a Zip Deploy

Existing build process

A lot of work to rebuild in Azure

A connector that joins the output of your build pipeline to Azure

Demo

Look at a brief script I’ve prepared for Zip deployment

Understand each line

Deploy and verify a change in the code

Making This Work with Your Process

Zip the output of your build

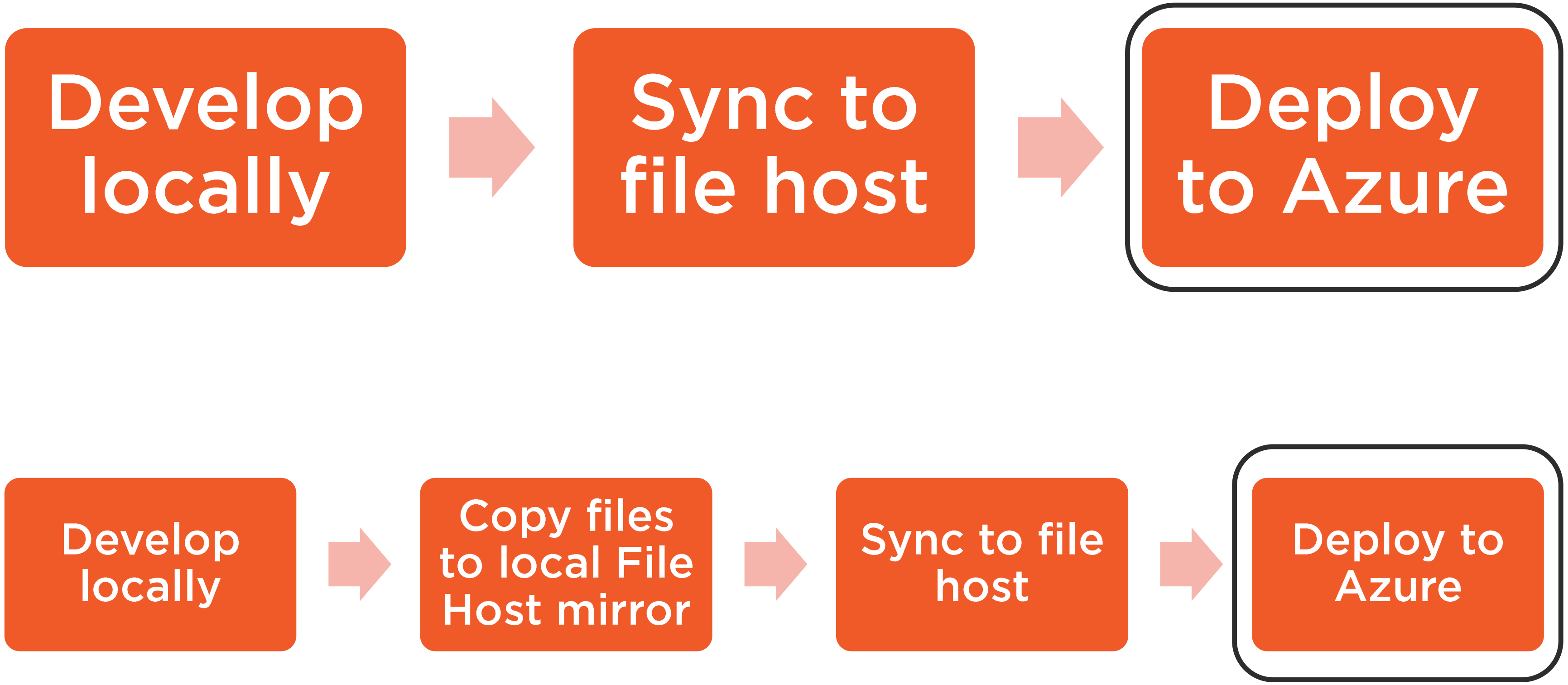
(through the zipdeploy script if you prefer)

Generalize what you need to in the script

(like the zip file location)

Execute zipdeploy.ps1 script at the end of your build

Deploying from a File Hosting Service



Develop locally

Sync to file host

Deploy to Azure

Develop locally

Copy files to local File Host mirror

Sync to file host

Deploy to Azure

Demo

Configure Azure to trust our Dropbox, and vice versa

Configure Dropbox as a deployment source

Make a small change to the web application

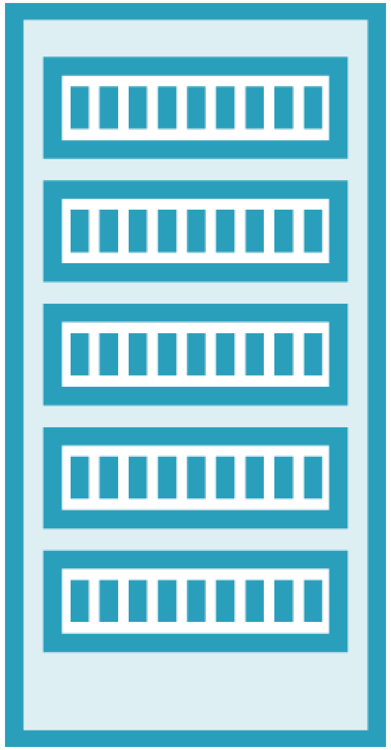
Deploy and verify the change

Deploying with an ARM Template

Code is maintained and deployed

The target asset to which it’s being deployed is NOT

Managing Infrastructure



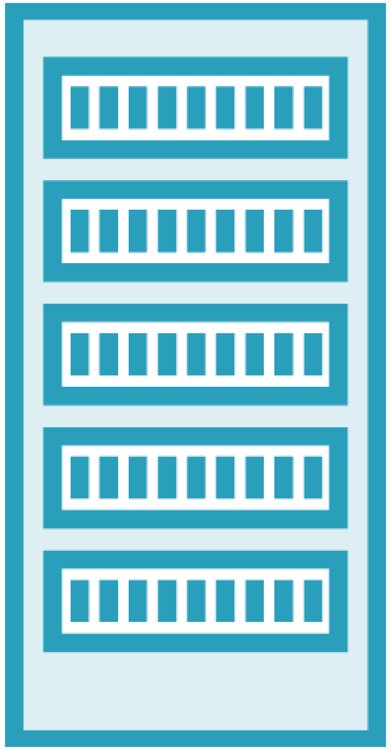
Infrastructure exists first as human-being actions

Re-provisioning prompts the move to script

- Which can be version controlled
- Scripts are prescriptive
    - *Create Web App*
- The alternative is DECLARATIVE
    - *There IS a Web App*

Infrastructure as Code

the process of managing through definition files, rather than physical hardware configuration or interactive configuration tools

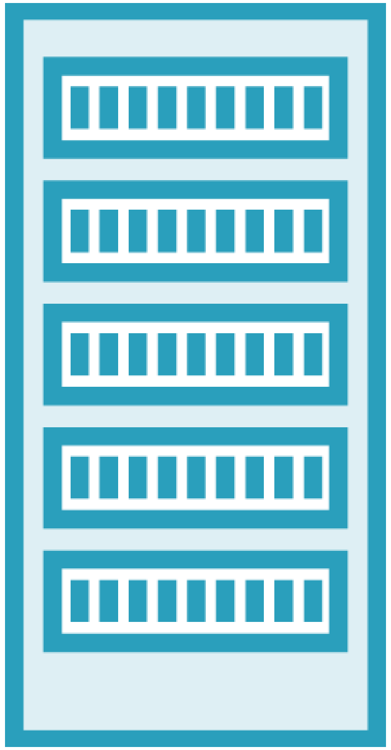


Azure

Resource

Management

Templates



Azure

Resource

Management

Templates

ARM Template Fast Facts

JSON files

Split into Subjects and Adjectives

Templates and Parameters files

ARM Templates are used internally by Azure in EVERY deployment

Whether YOU created the template or not

Demo

Capture the existing ARM Template for our Web app

Tweak the template so that we can use it for deployment

Make a simple change to the app settings in the template

Deploy and verify the change

Demo

Learn how our settings relate to the interface

Add a new default document to the app we just created

Add a Sql Server server and database using nothing but an ARM template

Excited about Infrastructure as Code?

Databases don’t do Infrastructure as Code

https://www.pluralsight.com/courses/deploying-databases-octopus

Changes in database state cannot be inferred declaratively

Database Migrations

Create a new nullable byte field to store the hashed password

Execute an update query against the new password field using the hashed value of the existing password field

Drop the original password field

Rename the new password field to have the same title as the old field

Database Migration Tools

RedGate's ReadyRoll

DbUp

Flyway

ARM Template Possibilities

Many use cases

Provisioning multiple environments

- Dev
- Staging
- Production

Provisioning Multi-tenant instances

Summary

Looked at all of the Azure Deployment options

Organized them into a map of the different categories

Performed a series of deployments

- Zip file
- FTP
- Dropbox
- Learned about ARM Templates

Performed some simple but useful deployments with them