Octopus Deploy

DevOps in two days with Github Actions, Azure, Infrastructure as Code, GitHub Actions and Octopus Deploy

By Derek Campbell

Day one - Morning Agenda - Sign-up

- Settling in, Computer setup, Introductions
- Prerequisites
- Workshop begins
- Github Sign-up
- Octopus Deploy sign-up
- Azure Sign-up

Pre-requisites

- VS Code
- MSBuild
- .NET Framework 6.0+
- Git Desktop/client
- Credit Card for Azure sign-up unless using existing account

https://github.com/OctopusDeployCommunity/NDCDevOpsWorkshop

Github Sign-up Demo

Q&A

Introducing Octopus Deploy

- Helps DevOps teams at over 25,000 companies accelerate reliable, repeatable and traceable deployments to cloud and on-Premises
- Includes 500+ automation step templates
- Integrates with hundreds of technologies
- Store your application code and Config as Code in the same repository

Octopus Cloud Sign-up Demo

Q&A

Microsoft Azure

- Azure is a highly-scalable public computing platform
- Azure supports:
  - IaaS
  - PaaS
  - SaaS
- Azure has 52 regions with 6 planned
- Azure works in region pairs - West US & East US
- Offers Availability zones for highly scalable apps with Zone redundancy


Norway West
Norway East
West Europe
Germany West Central (Public)
UK South
Germany North (Public)
North Europe
Germany Northeast (Sovereign)
UK West
Germany Central (Sovereign)
France Central
Switzerland North
France South
Switzerland West
Spain Central
US Gov Iowa
Central US
Canada East
West US 2
Canada Central
West Central US
North Central US
West US
US DoD East
US Gov Arizona
East US
East US 2
South Central US
US Gov Texas
US Gov Virginia
Mexico Central
US DoD Central
Israel Central
Qatar Central
UAE North
UAE Central
West India
Central India
South India
Southeast Asia
China North
China North 2
Korea Central
Korea South
Japan East
Japan West
China East 2
China East
East Asia
Brazil South
South Africa North
South Africa West
Australia East
Australia Central
Australia Southeast
Australia Central 2
Available region
Announced region
Availability Zones

Microsoft Azure sign-up Demo

Q&A

End of Morning session

Day one - Afternoon - Github Actions & Octopus Configuration

- Recap of morning session
- Azure Infrastructure
- Azure Service Principal Set-up
- Github Actions Configuration
- Octopus Deployment setup

Infrastructure as Code

“Infrastructure as Code (IaC) is the management of infrastructure (networks, virtual machines, load balancers, and connection topology) in a descriptive model, using the same versioning as DevOps team uses for source code. Like the principle that the same source code generates the same binary, an IaC model generates the same environment every time it is applied. IaC is a key DevOps practice and is used in conjunction with continuous delivery.”

https://docs.microsoft.com/en-us/devops/deliver/what-is-infrastructure-as-code

Azure Service Principals

“An Azure service principal is an identity created for use with applications, hosted services, and automated tools to access Azure resources. This access is restricted by the roles assigned to the service principal, giving you control over which resources can be accessed and at which level. For security reasons, it's always recommended to use service principals with automated tools rather than allowing them to log in with a user identity.”

Azure Service Principal Setup

Q&A

Introducing Github Actions

- 13,000+ Actions on one of the largest DevOps ecosystems
- Seamless brings community standards into development
- Home for the world's code, including Octopus Deploy
- Natively integrated with Github & CodeSpaces
- Accelerate with your preferred language and environment

Github Actions Demo

Q&A

Introducing Octopus Deployments

- Projects let you create and manage your deployment processes, releases and Runbooks.
- For each project you define you deployment process
- You can manage these in the Octopus UI or using the Rest Client
- You can use Config as Code to manage your deployment process. (More on that later)

Octopus Deploy Project Configuration Demo

Q&A

End of Day One

Day two - Morning - Octopus Runbooks & Execution

- Recap of Day One
- Octopus Runbooks setup
- Execution
- Troubleshooting

What are Octopus Runbooks?

- Runbooks are the “Ops” side of DevOps
- Runbooks are used to automate routine configuration, maintenance, and emergency operations tasks
- Infrastructure provisioning, Dependency installation, Turning Infrastructure On/Off
- Database management, server updates
- Website/DR failover, Restore from backup, Spin up DR, Restarting Services

Why use Octopus Runbooks for IaC?

- Octopus Runbooks is built for Infrastructure as Code, routine and emergency tasks
- Centralized deployment of Infrastructure from a shared platform
- Use built-in variables as part of your IaC naming conventions
- Deploy your Infrastructure, application and databases all in a single pane of glass
- Spin up and tear down on a schedule

What problems does IaC solve?

- Built to solve environment “drift” in the release pipeline
- It avoids treating each environment as “snowflakes”
- Idempotence
- Enables teams to test applications in production-like environments early on in the life cycle
- Allows the IaC to be treated like application code

IaC on Azure

- Azure provides native support for IaC via Azure Resource Manager
- Teams can define templates that specify the infrastructure
- Azure supports:
  - ARM Templates
  - Bicep
  - Terraform
  - Pulumi

What are ARM Templates?

- ARM Templates allow you to create entire Azure Infrastructure stacks declaratively.
- Infrastructure deployments that can be deployed concurrently
- Consists of:
  - Parameters
  - Variables
  - User-defined functions
  - Resources
  - Outputs

What are the benefits of ARM?

- Repeatable results
- Don’t need to worry about the complexity of ordering operations.
- Modular reusable files
- Built-in validation
- Tracked Infrastructure deployments
- CI/CD integration
- Policy as Code
- Allows you to create any Azure resource
- Extensible with PowerShell or Bash

Octopus Runbooks & ARM Demo

Q&A

Execution

- Build Infrastructure using Runbooks
- Run Github Action
- This should complete the deployment to Production

Workshop Execution Demo

Q&A & Troubleshooting

Day two - Afternoon - Octopus Runbooks & Execution

- Recap
- Octopus Configuration as Code
- Additional Information
- Wrap-up

What is Config as Code

- Store deployment processes in code
- Branch your configuration and test changes in the branch before merging them
- Review and collaborate on changes using pull requests
- Clone an existing project to use as a template for future projects

What is Config as Code

- Track changes to the deployment configuration using the same tools you already use for your application code
- Edit your deployment configuration in your preferred text editor or in the Octopus app
- Work where you work best

What is Config as Code

- The configuration is stored as human-readable files that use Octopus Configuration Language (OCL). We designed OCL to make it easier to read and edit the deployment process and review any changes.
- VSCode extension on marketplace for OCL (Demo later)

What is Config as Code

- Not everything is moved into the repository when you enable version control. A list of version-controlled resources is available in our Configuration as Code reference.

What's included in Config as Code

- Deployment Process
- Deployment Settings
- Release Versioning
- Release Notes Template
- Deployment Targets Required
- Transient Deployment Targets
- Deployment Changes Template
- Default Failure Mode

What's stored in the Octopus Database

- Channels
- Triggers
- Releases
- Deployments
- Runbooks
- Variables
- General Settings
  - Project Name
  - Enabled / Disabled
  - Logo
  - Description
  - Project Group

Resources NOT version controlled

- Infrastructure
  - Environments
  - Deployment Targets
  - Workers
  - Worker Pools
  - Machine Policies
  - Machine Proxies
  - Accounts
- Tenants

Resources NOT version controlled

- Library
  - Certificates
  - External Feeds
  - Lifecycles
  - Packages
  - Build Information
  - Script Modules
  - Step Templates
  - Variable Sets

Resources NOT version controlled

- Server Configuration
  - Feature Flags
  - License
  - Node Settings (Task Cap and Server Uri)
  - Issue Tracker Settings
  - External Auth Provider Settings
  - SMTP
  - Spaces
  - Teams (both membership and role assignment)
  - Users
  - User Roles

Config as Code is not a replacement for the Terraform provider

Config as Code is not a way to sync multiple instances of Octopus Deploy

Config as Code Patterns

- Keep your Octopus configuration alongside your application code
- Branches as drafts, pull requests as approvals
- Branch-specific variables as a naming convention
- Leverage Development tools and practices where appropriate
- Leverage the UI when appropriate

How to use Config as Code

- Available as of 2022.1
- Converting an Octopus project is a one-way process
- We recommend that you clone any test projects using Project Import/Export
- Validate it works as expected
- Roll it out in stages across the company and implement standards
- Feedback is welcome

Config as Code Roadmap

- Released in EAP to Cloud in November 2021
- Deployment process is live as of 2022.1
- Variables (Cloud first, GA in Q3/4)
- Runbooks (Cloud first, GA in Q3/4)

Config as Code Demo

Q&A

Please provide feedback on https://oc.to/workshopfeedback

Thank you for attending

I hoped you learned something cool

https://octopus.com/