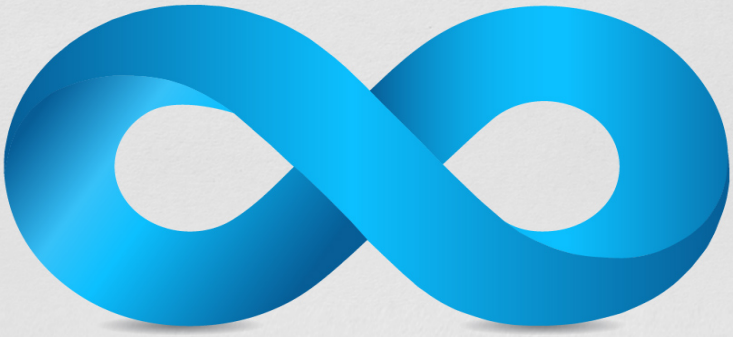
From Admin to DevOps



The No-BS Way to DevOps in Azure

ADAM BERTRAM, MICHAEL LEVAN

From Admin to DevOps

The No-BS Way to DevOps in Azure

Adam Bertram and Michael Levan

This book is for sale at http://leanpub.com/fromadmintodevops

This version was published on 2021-04-25

This is a Leanpub book. Leanpub empowers authors and publishers with the Lean Publishing process. Lean Publishing is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.

Contents

Preface

There is a lot of hype out there about the word 'DevOps'. It seems like every product vendor's marketing team is touting their 'DevOpsness' and every organization needs to have a 'DevOps Engineer' role. Cut past all of the hype though and it all eventually comes down to getting shit done by delivering value to the business. That 'value' comes in the form of both culture and tools.

One way to deliver value to businesses is the cloud. The cloud pours gas on the fire of any DevOps journey via it's no-hardware-to-manage, API and automation-driven environment. Nowadays, everything is software and code. And when you can build with code, you have the power to automate just about anything.

There are a few major public cloud providers out there but Azure has gained a lot of traction in the past few years. It has invested billions of dollars building its Azure cloud platform and the market has taken notice. Along the way, they have been continually building out a robust tooling ecosystem with their Azure DevOps service and especially with their GitHub acquisition.

Microsoft should continue to be a strong contender in the race for DevOps in the cloud.

If you learn anything from this book, we hope that you will come away with a sense of understanding there's no one way to implement DevOps. However, we encourage you to follow along with all of the projects you'll learn in this book. Occasionally, we encourage you to take a step back from the tactical do this, do that project approach and realize the bigger picture from time to time.

DevOps isn't just a new way of building and delivering software; it's a new mindset.

About the Book

This book's mission is to take a system administrator, IT professional or software developer from admin to DevOps. This book's primary goal is to introduce DevOps concepts like configuration management, continuous integration/continuous delivery, infrastructure as code and more. The aim is to show you, the reader, what's possible through real projects you'll see in the wild.

In this book, the authors will not cover much of the cultural aspect of DevOps. There are already many books out there that cover this area of DevOps well like *Building a DevOps Culture* by Mandi Walls[1], *Effective DevOps: Building a Culture of Collaboration, Affinity, and Tooling at Scale*[2] by Jennifer Davis, *The DevOps Handbook*[3] by various authors and *The Phoenix Project*[4] by Gene Kim.

Book Outline

The book is split into two parts, Tools and your DevOps journey.

Part I: Tools

In Part I: Tools, you will learn, hands-on, how to get started with over a dozen popular tools to establish a strong DevOps presence in your organization. These tools include:

1. Git and GitHub
2. Azure Boards
3. Azure Pipelines
4. Ansible
5. Terraform
6. Kubernetes
7. Docker
8. ...and more!

Each chapter in Part I will provide an excellent understanding of each tool *and* also how to use it by building a real project. Part I chapters will prepare you for what's to come in Part II.

[1] https://amzn.to/3dd5KRI
[2] https://amzn.to/3baYVOK
[3] https://amzn.to/2UouNsJ
[4] https://amzn.to/2UgzKUe

Part II: The Journey

Part II is where your DevOps journey begins. Part II is broken into milestones. This part assumes you're in your DevOps infancy and proceeds to teach you, hands-on, build, deploy and eventually maintain a real-world cloud application.

Part II is broken up into journey milestones with various chapters based on a real project. Basing the entire journey from the eyes of a team and an example project, you will follow the progress of an organization to:

1. Implement continuous integration (CI) with GitHub and Azure DevOps.
2. Implement continuous delivery/deployment (CD) with ARM templates, Terraform and Azure Virtual Machines.
3. Learn how to monitor Azure cloud infrastructure and application with Azure Monitor and Application Insights.
4. Take DevOps tooling to the next level with Docker, Kubernetes, and serverless technologies

By the time you're done working side-by-side on a real project with the authors in Part II, you will have the skills to be successful at any DevOps project in Azure.

Command Line Over the GUI

The tools and services you'll be learning in this book can be built many different ways. Since a primary component of DevOps is automation, expect a lot of command-line usage in this book.

Even though running commands from the command-line isn't automation, it's a great start. Throw together a few commands in a script and voila! You have a great start at an automation routine.

Using GUI tools like the Azure Portal and other web-based frontends is fine for getting your feet wet to discover what's possible but it doesn't work if you're 'doing the DevOps'. Even when building proof of concept (PoC) projects like you will in this book, use the CLI.

Always use the CLI where possible and save your work in scripts.

Why Azure? Why aren't you covering *my* tool?

DevOps applies to all kinds of situations and tools. We gotta start somewhere. Azure just so happens to be the authors' primary cloud of choice. You definitely don't *have* to do the DevOps in Azure but you definitely can…

This book was written to take a specific look at DevOps particularly in the Microsoft Azure space. DevOps, itself, isn't dependent on any certain environment whether it be on-prem or in the cloud. But, you'll find that the API-driven, automation and lack of hardware to manage aspect of the cloud will greatly accelerate you in your DevOps journey.

About the Authors

Adam Bertram is a 20+ year veteran of IT and an experienced online business professional. He's a consultant, Microsoft MVP, blogger, trainer, published author and content marketer for multiple technology companies. Catch up on Adam's articles at adamtheautomator.com[5], subscribe to the ATA email list[6] and follow him[7] on Twitter at @adbertram.

—

[5] http://adamtheautomator.com/
[6] https://mailchi.mp/adamtheautomator/zu5a0hlixv
[7] https://twitter.com/adbertram

Michael Levan is a tech enthusiast and engineer at heart. He is a Developer Advocate at Octopus Deploy and is the Founder & Chief Engineer at CloudDev.Engineering LLC. Michael is a blogger, public speaker, author, all-around content developer, and utterly fascinated with DevOps and Cloud Engineering. Want to see what he is up to? clouddev.engineering[8] or Twitter @TheNJDevOpsGuy[9].

[8] http://clouddev.engineering
[9] https://twitter.com/thenjdevopsguy

Target Audience

This book is for anyone in IT or software engineering that wants to learn more about DevOps and, specifically, how to follow DevOps practices in Azure. Whether you're a system administrator, engineer, developer or IT manager, this book will help you understand what it takes to deliver more value to customers with DevOps using Azure.

The book is especially suited for system administrators (admins) that have suddenly found themselves needing to 'do the DevOps'. If you're in a position within your company where you've been managing the operations side for a long time but aren't quite sure about this DevOps thing, this book is for you.

If you think you have a beginner or intermediate-level of knowledge on DevOps concepts and are interested in seeing how DevOps can be applied in Azure, this book will help.

Still not sure? The authors have built a persona to base the entire book around. If you find yourself nodding your hand in agreement with many of these statements, you need this book.

You

- You've been in IT for 5+ years
- You've always been a sysadmin and mostly managed Windows servers but a few Linux boxes here and there
- You've enjoyed writing scripts in the past few years automating random stuff. Mostly PowerShell and Azure CLI.
- You've heard of DevOps and have just begun to play with DevOps tooling like Ansible, DSC, etc. This stuff is pretty interesting.
- You learn best with examples and real projects.

Your Organization

- Your organization is medium-sized and has a few dev teams with a QA dept for all working on 2-3 in-house applications.
- You have some automation to deploy software to production but it's a combination of scripts and other tools I didn't implement.
- You have complete management buy-in for DevOps and automation.

Your Goals

- You want to learn how to ‘do the DevOps’ like your manager has been telling you. He’s been seeing all this marketing stuff about ‘DevOps’ and he’s sold.
- You want to get a grasp on exactly what the hell DevOps is and wrap my head around it
- You’ve read high-level DevOps books and have also been reading product-specific blog posts about small things but you’d like to see an end-to-end example.

Code Samples and Resources

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

The BookResources GitHub Repository

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

The NoBSAzure GitHub Repository

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Part I: Tools

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Understanding the Tool

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

The Project

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Azure from the Command Line with Cloud Shell

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

The Azure Cloud Shell

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Setting up the Cloud Shell for the First Time

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Using the Cloud Shell

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

The Azure CLI

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

No Setup Needed using Cloud Shell

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Commands and Arguments

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Command Groups

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Finding Commands

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Finding Examples

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Getting Interactive

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Extensions

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Idempotence

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Scripting

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Azure PowerShell

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

No Setup Needed using Cloud Shell

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Finding Commands

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

The Verb-Noun Syntax of PowerShell Commands

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Finding Help

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Conclusion

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Resources

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Visual Studio Code

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

VS Code: An Overview

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

The Interface

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Commands and the Command Palette

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Settings

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Extensions

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Snippets

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

The Integrated Terminal

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Working as a Team with Workspaces

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Workspace and Settings

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Building Infrastructure Without Leaving VS Code

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Project Overview

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Tools to Have

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Setting up a Workspace

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Clone the Git Repo

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Creating a Workspace

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Setting up Extensions

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Code Editing

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Saving Time with Snippets

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Commit Code to Git

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Running Commands in the Integrated Terminal

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Resources

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Azure Boards

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Azure Boards: The Basics

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Work Items

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Backlogs

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Sprints (Iterations)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Kanban Boards

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Building and Organizing a Sprint

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Tools to Have

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Assigning Team Members

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Adding Team Members the Project

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Creating the Sprint (Iteration)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Set Sprint Capacity

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Building the Backlog

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Creating a User Story

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Creating Tasks and Bugs

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Summary

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Resources

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Git and GitHub

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

How Git and GitHub Work Together

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Collaborating Outside of Your Team

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Branching Basics

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

An Example Use Case

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

The GitFlow Branching Strategy

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Merging Code

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Pull Requests

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Visual Studio Code (VS Code) + Git/GitHub

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Building Your First Collaborative GitHub Repository

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Project Overview

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Tools to Have

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Installing Git

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Creating a GitHub Repository

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Adding Collaborators

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Cloning the Repository

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Staging and Committing Code

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Creating a Develop Branch Using VS Code

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Creating a Pull Request and Merging Into Master

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Reviewing and Merging a Pull Request

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Summary

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Resources

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Azure VM Availability Sets

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Azure VM Fault and Update Domains

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Availability Sets

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Availability Set Design

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Creating and Testing an Availability Set with Terraform

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Project Overview

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Tools to Have

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Building the Terraform Configuration

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Inspecting the Resources in the Azure Portal

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Confirming Results

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Resources

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Infrastructure as Code

If you've ever created anything in Azure, you'd know there's either a lot of clicking or a lot of typing going on. To provision anything in Azure requires remembering which screen to go to or what command to run. If you're just playing around trying to learn Azure that's fine but once that playing turns into business-critical, production processes where time is money, something has to change.

Once your organization starts to get serious about cloud infrastructure, you're going to be facing a lot of new challenges. You'll encounter unnecessary duplication of work, repeated cases of fat-finger syndrome, change management issues, configuration drift[10] and more. The larger the organization and team, the larger the problems.

You'll see a full breakdown of how Azure handles this challenge in the next chapter. But, before we can get into ARM templates, it's important to first understand an umbrella term that ARM templates fall under called Infrastructure as Code (IaC)[11].

Infrastructure as Code

IaC is an industry term that refers to storing all of the stuff required to build infrastructure components in code. That code is typically defined in JSON or YAML files representing what your infrastructure *should* look like.

Once all components are defined in a structured file, another process comes along, understands the code and immediately begins using that document as instructions to build the infrastructure.

To provide a generic, pseudocode-like example, perhaps you have a VM to create. That VM has needs like compute, storage and networking. A rough IaC approach would be to describe the VM and all of its components in a template.

The Template

In the following code snippet, you can see an example of how each component could be broken down by a hierarchy with attributes defined about each component. This example was created in a JSON file most services would call a *template*. In the Azure world, this fictional template is an ARM template.

[10] https://dzone.com/articles/configuration-drift
[11] https://docs.microsoft.com/en-us/azure/devops/learn/what-is-infrastructure-as-code

```
{
    "VM": {
        "Name": "MYVM",
        "Disk": {
            "Size": "100GB"
        },
        "Networking": {
            "IPAddress" : "10.0.0.1"
        }
    }
}
```

This template defines the VM and all of the attributes associated with that VM. It has a specific *schema* that template authors adhere to define what that VM looks like. A schema file is defined as a URL in the template which, when deployed, downloads and compares the template with the schema file.

Suppose this template is then saved to a file called *myvm.json*.

Source Control

You now have everything that makes up a VM saved in a single file. Like any good DevOps professional, you check that file in source control. You now have a way of tracking changes to the file.

The Tool

Now that the file is created, you need a tool or service to read that file that understands what you're trying to build. That tool uses the template as input and builds the VM to those exact specifications with no other interaction.

```
[some command line tool] -File myvm.json
```

Good deal, right? That's not all.

Fighting Configuration Drift

Now suppose you need to change the static IP address assigned to that VM's NIC. You *could* RDP to the VM and change the IP but you don't want to do that. Why?

1. You're making the change manually which wastes time and is prone to human error.
2. There's no audit trail of who changed the IP and when.

3. There's no automated way to revert the change should you fat-finger the IP address.

IaC can alleviate all of the above challenges. With, literally, a few keystrokes, you could make that change in a way your manager and auditor will love you for.

Open up *myvm.json,* change the `IPAddress` attribute, commit the change to source control and run the tool again. Done.

```
{
    "VM": {
        "Name": "MYVM",
        "Disk": {
            "Size": "100GB"
        },
        "Networking": {
            "IPAddress" : "10.0.0.2"
        }
    }
}
```

```
[some command line tool] -File myvm.json
```

The tool will be smart enough to know what needs to change. It won't detach the NIC or rebuild the entire VM. All IaC tools and services are smart enough to understand how to make the change. Magic!

But now you and your just realized you used the wrong IP and need to revert back. No problem. Revert the change in source control, commit, run the tool and you're back in business.

But wait, there's more.

The Beginnings of Continuous Delivery

When you have infrastructure stored in templates under source control, you have a set of ingredients for the start of an automated release or continuous delivery[12] pipeline.

Recall that you had to run that tool every time you changed the template. In an automated pipeline/workflow, that tool runs automatically. Once the process to create or make the change to the environment is automated, the moment you commit a change to the template, the infrastructure matches.

Build enough templates and eventually, your entire infrastructure could be represented in code or *as code.*

[12] https://continuousdelivery.com/

Azure Resource Manager (ARM) Templates

ARM Templates: Infrastructure as Code

The Schema

Use Examples and Syntax "Helpers"

Deploying an Azure Virtual Machine (VM)

Project Overview

Tools to Have

Downloading an Azure Quickstart Template

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Inspecting the ARM Template

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Inspecting the Parameters File

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Creating the Resource Group

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Deploying the ARM Template

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Cleaning Up

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Resources

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Terraform

Have you ever tried to create a ton of infrastructure resources on-prem or in Azure manually like virtual machines (VMs) or really any kind of Azure resource with the Azure portal?

There are three primary problems of manually creating resources:

- The task is error-prone. As humans, we make mistakes. Pointing and clicking in a UI opens up a whole new world of making mistakes because it's incredibly simple to click a wrong button or type in a wrong IP address.
- The task is cumbersome. Creating multiple resources manually is simply annoying and takes forever.
- Even if you create the right components now, configuration drift[13] is bound to happen. If you ask three engineers to create two virtual machines, chances are all three will create the VMs differently. Some may use the CLI and others may use the UI. This approach opens up opportunities for non-standard configuration.

How do we resolve these problems? We write code, specifically, Terraform code.

Terraform is an Infrastructure-as-Code (IaS), immutable[14] programming language created by Hashicorp[15] that defines infrastructure components whether it be on-prem or in the cloud, as code.

Code isn't cumbersome and if it's written correctly in a reusable way, chances are it only has to be done once. Since the code is being used across the environment, you don't have to worry about configuration drift because the code doesn't change.

Many IaS tools exist already ranging from open-source, closed-source, and vendor-specific. Each IaS tool has its pros and cons, but Terraform is quickly becoming the standard in many Azure deployments. In fact, Microsoft is basing a ton of documentation[16] off of Terraform.

Terraform simplifies the way that you can deploy Azure resources. In this chapter, you will learn the core fundamentals of Terraform to prepare you to take the experience you've learned and start building Azure resource with Terraform.

Configuration Files

One of the most important concepts to understand in Terraform is the concept of a configuration file. Terraform configuration files (files with the TF extension) are the foundational component of

[13] https://www.continuitysoftware.com/blog/it-resilience/what-is-configuration-drift/
[14] https://medium.com/tribalscale/understanding-immutability-fdd627b66e58
[15] https://www.hashicorp.com/
[16] https://docs.microsoft.com/en-us/azure/terraform/terraform-overview

Terraform. Configuration files provide instructions to the Terraform binary on what infrastructure to build and how to build it. These configuration files ultimately become a Terraform *configuration*.

Terraform has several configuration files, which you will see throughout this chapter and Part 2 of the book. The main (pun intended) configuration file is called *main.tf*. The *main.tf* configuration file holds what resources or services in Azure you will be creating.

Although you can call the "main" configuration file something other than main.tf, it's common practice to call it main.tf.

For example, perhaps you are creating a virtual network. The code to create the virtual network resource will be stored in the *main.tf* configuration file. You can see below what a *main.tf* configuration file may look like.

```
resource "azure_virtual_network" "default" {
  name          = "test-network"
  address_space = ["10.1.2.0/24"]
  location      = "West US"

  subnet {
    name           = "subnet1"
    address_prefix = "10.1.2.0/25"
  }
}
```

Source: https://www.terraform.io/docs/providers/azure/r/virtual_network.html

Under the `resource` configuration file to create a virtual network, you will see *arguments*. Arguments contain values like the name of the virtual network, the location or region of the virtual network, IP addresses associated with the virtual network, and many other components depending on what type of resource you are creating.

Hashicorp Configuration Language (HCL)

In a sense, Terraform *is* code. Where there's code, there's a language. Sometimes that language is a programming language and other times it's some sort of data serialization[17] language. All Terraform configuration files are written in a language called HCL.

[17] https://devopedia.org/data-serialization

HCL code gets stored in configuration files and the configuration files are defined using Terraform configurations. When you run Terraform configuration files (you will learn more about running Terraform configuration files in later sections), the code inside of the configuration files will be triggered to create Terraform resources.

HCL is a programming language created by Hashicorp to be both human-readable and for a machine to quickly compile. HCL's original design was to be geared towards DevOps tooling around automation of infrastructure and services, which still very-much holds true.

```
name                = var.Name
location            = var.location
resource_group_name = var.resourceGroup
dns_prefix          = "[redacted]"

default_node_pool {
  name = "default"
  node_count = 1
  vm_size = "Standard_D2_v2"
}
```

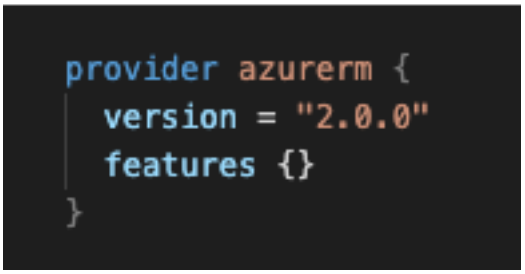
HCL Example

You may notice that HCL configurations look very much like JSON, or key-value pairs. Hashicorp made HCL in such a way that it is fully compatible with JSON and JSON can be used as an input system

Providers and Resources

Terraform isn't just a tool to deploy Azure resources, it can also work with on-prem, AWS and other providers. To support each of those providers, Terraform has a concept called *providers*. Think of a Terraform provider[18] like the medium between Terraform and Azure. For Terraform to communicate with Azure, it needs a path to get there. It needs to understand how Azure works. Terraform creates and maintains this "path". But how does the provider know how to communicate with Azure?

[18] https://www.terraform.io/docs/providers/index.html



Example Azure provider in a Terraform configuration file

Hashicorp builds the Azure provider (and all other providers for that matter) in a way for Terraform to make API calls to Azure. Because the provider is conducting API calls, Hashicorp needs access to Azure backend resources like virtual machines, web apps, serverless, etc. Once Hashicorp receives access from Microsoft, they are then able to start creating resources against that particular provider.

Resources[19] are how Terraform creates services and infrastructure. For example, to create a virtual machine using Terraform, you would use the `azurerm_virtual_machine` resource. The resource is comprised of several parameters that are passed in when you create the virtual machine. These parameters include things like the virtual machine name, what resource group the virtual machine will reside in, and so on.

Input Variables (Variables)

Variables in Terraform are like in any other programming or scripting language - they hold values. Variables can hold many different types of information to call on later in a configuration file, for example.

Variable Definition Syntax

Variables are a handy HCL constructs that can be defined a few different ways in a few different places.

Blocks

Variables can be defined in a block with various arguments in the form:

[19] https://www.terraform.io/docs/extend/resources/index.html

```
variable "<variable name>" {
    type = <type> ## string, number or bool
    <default = <value>>
    <description = "">
}
```

You can find more information about each type of argument assigned to a variable in the Input Variables Terraform documentation[20].

Let's say you're creating a configuration to provision an Azure VM from scratch. Inside of that configuration, you are referencing the VM name in various places. To ensure you only have a single place to change if that VM name should change, you've decided to create a variable.

The below example defines a variable called `virtualMachineName` and is assigning it a default value of `NoBSVM`. Since `NoBSVM` is text, the variable sets a `type` of string.

```
variable "virtualMachineName" {
        type = string
        default = "NoBSVM"
}
```

Assigning default values to variables isn't necessarily best practice. Why? Because you are statically assigning a value to the variable that is tied to a specific name.

"Simple" Syntax

The above example of defining an Azure VM name created a variable called `virtualMachineName` and assigned it a value of `NoBSVM` in a block with four lines of code. This approach wasn't entirely necessary. Since the variable definition was a simple string, it could have been represented in "shorthand" syntax like below.

```
variable "virtualMachineName" = "NoBSVM"
```

You can see that this approach saves a little typing yet performs the same function.

At Runtime

Whenever Terraform runs, you can also create variables without defining them in a file at all. When you run the Terraform binary, you can add an additional argument creating a variable on the fly that is then available in the configuration.

All variables that are run from the command-line start with the `-var` switch. Following the switch, there is an `=` sign to define the variable and a key-value pair representing the variable name and value. You can see an example of the syntax on defining a `virtualMachineName` variable at runtime below.

[20] https://www.terraform.io/docs/configuration/variables.html

```
> terraform apply -var="virtualMachineName=NoBS-VM"
```

Variable Definition Locations

Now that you know *how* to define variables, you should now know *where* to define them. In Terraform, you have two different locations to choose from.

- In the main configuration file (*main.tf*)
- In a separate configuration file e.g. *variables.tf*

In the Main Configuration File

If you have a simple configuration that doesn't require anything fancy, you can define variables directly in the main configuration file. Although, it's typically not recommended to do this due future code management concerns, it's perfectly OK to do it if you're just testing or working with simple configurations.

You can define variables in the main configuration file by any syntax you've read above.

In a variables.tf Configuration File

Alternatively, and undoubtedly a better approach is defining variables in a separate configuration file. When you define variables inside of a separate configuration file, you split up the variable declarations from the main configuration file allowing you to more easily manage varables.

When defining variables in a separate configuration file, you will typically not define a default value. This approach allows you to pass in values to `virtualMachineName` at run-time.

```
variable "virtualMachineName" {
        type = string
}
```

Variable Definition Files (TFVARS)

Let's say you've already defined a variable in the *main.tf* configuration file or a variables configuration file. You've defined the type, description, or other metadata about that variable but you haven't actually assigned a value yet. The variable definition file is a great way to assign that value.

One of the best ways to define variables is with a TFVARS file or simply a file ending with the tfvars file extension.

Let's say you have already defined a `virtualMachineName` variable in a configuration file but haven't provided a default value and are not passing a value at run-time. You can assign that value in a variable definition file.

Within a variable definition file, you could assign the value as shown below. Let's say you've assigned the variable as shown below and have saved it in a file called *testing.tfvars*. When you run the configuration, Terraform will pick up this value by passing the file at run time with `terraform apply -var-file="testing.tfvars"`.

```
virtualMachineName = "NoBS-VM"
```

Referencing Variables

So far, you've learned a lot about how to *define* variables but ultimately you need *read* them too. Reading variables is the easy part! Once you've defined a variable and have assigned it a value, you can reference that variable's value by using an expression.

Perhaps you've created the variable called `virtualMachineName` and need to reference that variable's value inside of a configuration file. To do that, you can use an *expression*. You can see an example of the syntax to do this below.

```
resource azurerm_virtual_machine "VM" {
    name = var.virtualMachineName
}
```

Modules

So far, you've learned about various configuration files like the main configuration file (*main.tf*), an optional variable configuration, and a variable definition file (TFVARS). These three files are stored in a single directory. The directory is typically named after what the configuration as a whole is creating. For example, if the configuration creates an AzureVM, the directory might be called *AzureVM*. The directory (*AzureVM*) where the Terraform configuration files are stored is called a *module*[21].

State File

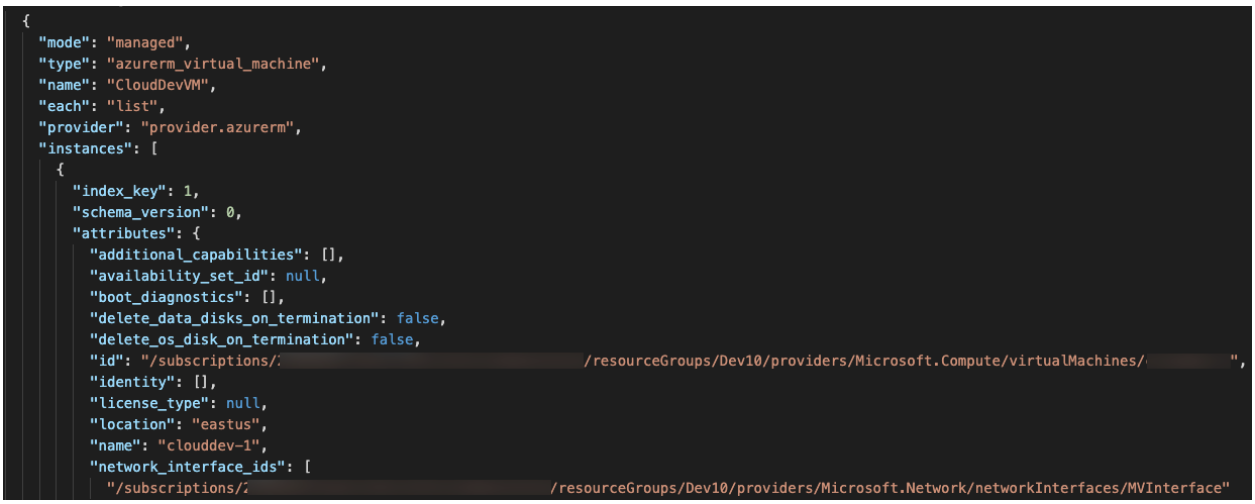
When you run the Terraform binary, Terraform reads all configuration files inside of the current directory to figure out what to build. It evaluates all of the variables and other logic you created inside of those configuration files to create a *state*. That state then represents the values of all configuration items Terraform needs to manage infrastructure defined in the configuration files or module.

A state file consists of all metadata for the resources Terraform is managing. The state file contains components like the subscription ID, virtual machine name, resource group that the resources will reside in, and many others.

[21] https://www.terraform.io/docs/modules/index.html#standard-module-structure

Think of a Terraform state file like a bucket of products to create chocolate chip cookies. You have chocolate chips, flour, mixing supplies, etc.. Without all of those products, you can't make chocolate chip cookies and without a Terraform state file, you can't create or manage Azure resources.

When you run Terraform, Terraform creates a *state file* to store the current state of the resources under management called *terraform.tfstate*. In the below example, the state file has holds what type of resource is being created, the provider that's being used, name, location, and so on. The state file holds *everything* Terraform needs to create or update those resources.



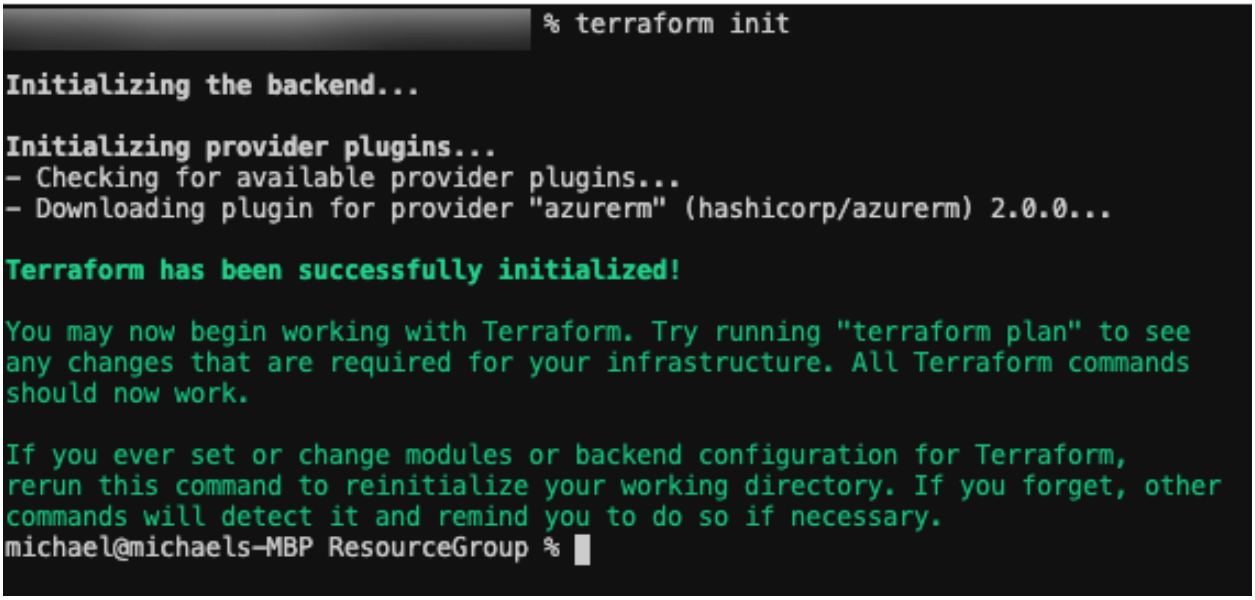
The Terraform Workflow (init —> plan —> apply)

In the previous section you learned all about variables, main configurations, and metadata in TFSTATE files. As is, you've just built a bunch of text files. The real goal here is to create some resources! You need to invoke Terraform to read the files and get stuff done!

Initialization

Let's assume you've already created some configuration files in a directory. Perhaps you've got a *main.tf*, a *variables.tf*[22] and a *terraform.tfvars* file inside there. You now need to tell Terraform that this directory has a set of configuration files in it. To do that, you will run `terraform init`. The `iniit` argument reads each configuration file and checks to see which providers you've chosen to use. It then downloads instructions for each provider.

[22] http://variables.tf



For example, if you've defined the `azurerm` provider in a configuration file as shown below, you will then see a file show up called *terraform-provider-azurerm_v2.0.0_x5* (or similar) in the same directory as your configuration files.

```
provider "azurerm" {
  version = "=2.0.0"
  features {}
}
```

Validation

After Terraform downloads all providers, it's time to test the configuration to see if there are any problems. At this point, Terraform checks for potential problems like syntax errors, resource errors, authentication errors, variable errors, and runtime bugs. Terraform validates the configuration to ensure no issues arise once it actually begins to carry out the execution of the configuration.

To validate or *plan* the configuration, run the `terraform plan` command. `terraform plan` will refresh the state file with the newly added resources as well as plan said resources.

```
                                   % terraform plan
Refreshing Terraform state in-memory prior to plan...
The refreshed state will be used to calculate this plan, but will not be
persisted to local or remote state storage.


------------------------------------------------------------------------

An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  # azurerm_resource_group.DevRG will be created
  + resource "azurerm_resource_group" "DevRG" {
      + id       = (known after apply)
      + location = "eastus"
      + name     = "Dev2"
      + tags     = {
          + "environment" = "Dev"
        }
    }

Plan: 1 to add, 0 to change, 0 to destroy.
```

Build

After Terraform validates the configuration, it's time to actually run the configuration building (or updating) any resources in your configuration. Run `terraform apply`. You will see similar output to the `terraform plan`, except with one difference - an "are you sure?" confirmation. Once you're sure, type "yes" to proceed.

```
[redacted] % terraform apply

An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  # azurerm_resource_group.DevRG will be created
  + resource "azurerm_resource_group" "DevRG" {
      + id       = (known after apply)
      + location = "eastus"
      + name     = "Dev2"
      + tags     = {
          + "environment" = "Dev"
        }
    }

Plan: 1 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value:
```

Once you type “yes”, sit back and watch Terraform do all of the hard work.

```
  Enter a value: yes

azurerm_resource_group.DevRG: Creating...
azurerm_resource_group.DevRG: Creation complete after 1s [id=/subscriptions/[redacted]/resourceGroups/Testing92]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
```

Now that you understand the syntax, configuration files, and how to create a Terraform configuration, let’s jump into the chapter project!

Creating a Virtual Network with Terraform

Now that you’ve gone through the main components of Terraform, how to get up and running, and the concepts of what each Terraform configuration is, let’s take that knowledge and put it to action by creating a Terraform module!

Project Overview

In this chapter project you will create a virtual network (vNet) using the following Terraform concepts:

- A *main.tf*[23] configuration file
- A *variables.tf* configuration file
- A *terraform.tfvars* variable definition file

Once the configuration is created, you will then initialize, validate and finally deploy the Azure vNet.

Tools to Have

To follow along with the project in this chapter, please be sure that you have:

- Windows 10
- The Azure CLI[24] and you are authenticated to Azure[25]. You can optionally use the Azure Cloud Shell as well.
- Visual Studio Code (VS Code)[26]
- Chocolatey, the Windows package manager[27]
- Terraform VS Code extension[28] (optional)

Installing Terraform

You'll first need to install Terraform. There are a few ways to install Terraform on Windows. One of the easiest ways is to use Chocolatey. To install Terraform with Chocolatey, open up a command prompt or PowerShell console as an administrator and run `choco install terraform`. Once complete, Terraform should be installed.

[23] http://main.tf
[24] https://docs.microsoft.com/en-us/cli/azure/install-azure-cli-windows
[25] https://docs.microsoft.com/en-us/cli/azure/authenticate-azure-cli
[26] https://code.visualstudio.com/download
[27] https://chocolatey.org/install
[28] https://marketplace.visualstudio.com/items?itemName=mauve.terraform

```
PS C:\> choco install terraform
Chocolatey v0.10.15
Installing the following packages:
terraform
By installing you accept licenses for the packages.
Progress: Downloading terraform 0.12.24... 100%

terraform v0.12.24 [Approved]
terraform package files install completed. Performing other installation steps.
The package terraform wants to run 'chocolateyInstall.ps1'.
Note: If you don't run this script, the installation will fail.
Note: To confirm automatically next time, use '-y' or consider:
choco feature enable -n allowGlobalConfirmation
Do you want to run the script?([Y]es/[A]ll - yes to all/[N]o/[P]rint): A

Removing old terraform plugins
Downloading terraform 64 bit
```

Prepping the Configuration Directory

Before creating the Terraform configuration files, you need to have a single directory to store the configuration files. This project will be using a directory called *vnet*.

1. Create the *vnet* directory. This directory can be anywhere you'd like.
2. Start VS Code open the *vnet* directory. All Terraform configuration files will be created in this directory.

Building the Variable Configuration File

Although it's tempting to begin building the main configuration file first, it's usually best to create the variable configuration file (if you have one) first. You'll typically want to build this file first to know how the environment will be defined.

1. In VS Code, create a new file and save it as *variables.tf*.
2. Open the *variables.tf* configuration file and copy/paste the below content or download it from the BookResources GitHub repo[29].

[29] https://github.com/NoBSDevOps/BookResources/blob/master/Part%20I:%20Tools/Terraform/variables.tf

```
## The Azure subscription/account ID
variable "subscriptionID" {
    type = string
    description = "Variable for our resource group"
}

## The resource group name that the new vNet will reside in.
variable "resourceGroupName" {
    type = string
    description = "name of resource group"
}

## The Azure region that the vNet will reside in
variable "location" {
    type = string
    description = "location of your resource group"
}
```

Building the Variable Definition File

After creating the variable configuration file, it's time to assign values to the variables. To do that, we'll use a variable definitions file.

1. Create a new file called *terraform.tfvars* and save it.
2. Copy and paste the code below into the variable definition file or download it[30]. Be sure to add in the subscription ID from the Azure subscription you are using and optionally change the `location` if another region is closer to you.

```
subscriptionID = ""
resourceGroupName = "NoBSDevOps"
location = "eastus"
```

Building the Main Configuration

You are now ready for the main event (pun intended)! You're now ready to build the *main.tf* configuration file. The main configuration file will contain all of the resources to create and the `azurerm` provider so that Terraform can communicate with Azure.

1. Create a new file and save it called *main.tf*.
2. Copy and paste the code below into the *main.tf* configuration file or download it[31].

[30] https://github.com/NoBSDevOps/BookResources/blob/master/Part%20I:%20Tools/Terraform/terraform.tfvars
[31] https://github.com/NoBSDevOps/BookResources/blob/master/Part%20I:%20Tools/Terraform/main.tf

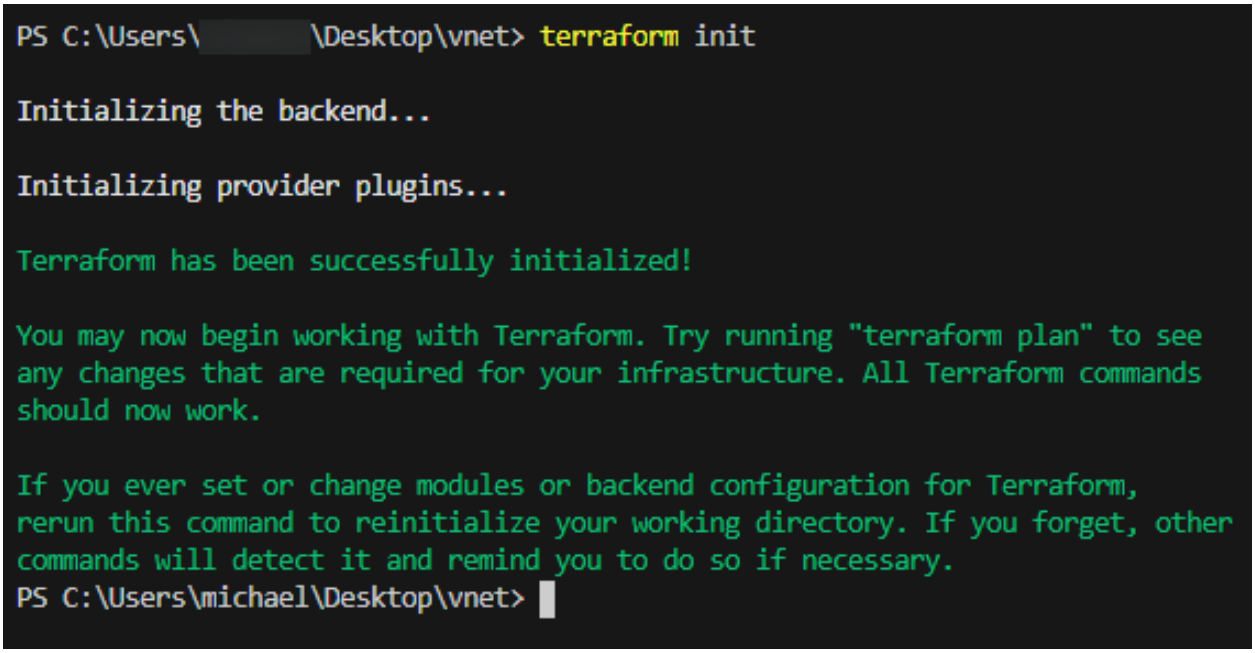
```
provider "azurerm" {
    version = "2.0.0"
    subscription_id = var.subscriptionID ## Reading value from terraform.tfvars

    features {}
}

## This resource creates the resource group to store all of the vNet resources in,
## like the network security group, for example.
resource "azurerm_resource_group" "NoBS" {
    name     = "NoBSDevOps"
    location = var.location ## Reading value from terraform.tfvars
}

## This resource is to create a network security group ([NSG](https://docs.microsoft\
.com/en-us/azure/virtual-network/security-overview)). Think of an NSG
## like a firewall.
resource "azurerm_network_security_group" "NoBSSG" {
    name                = "nobsSG"
    location            = "eastus"
    resource_group_name = var.resourceGroupName ## Reading value from terraform.tfva\
rs
}

## This resource creates the virtual network (vNet) itself. The primary
## parameter it contains is the address_space, which is the [CIDR](https://whatismyi\
paddress.com/cidr) range that
## will be associated with the vNet.
resource "azurerm_virtual_network" "NoBS-vnet" {
    name                = "nobs-vnet"
    location            = var.location ## Reading value from terraform.tfvars
    resource_group_name = var.resourceGroupName ## Reading value from terraform.tfva\
rs
    address_space       = ["10.0.0.0/16"]
    dns_servers         = ["8.8.8.8", "8.8.4.4"]

    tags = {
        environment = "Dev"
    }
}

## This resource creates the subnet that will be inside of the vNet for handing
## out IP addresses from a specific subnet range.
```

```
resource "azurerm_subnet" "nobs-sub" {
    name                 = "testsubnet"
    resource_group_name  = azurerm_network_security_group.NoBSSG.resource_group_name\

    virtual_network_name = azurerm_virtual_network.nobs-vnet.name
    address_prefix       = "10.0.1.0/24"

    tags = {
        environment = "staging"
    }
}
```

Deploying the Azure Resources

You now should have all of the requisites files created in the *vnet* directory. It's time to put Terraform to work using them.

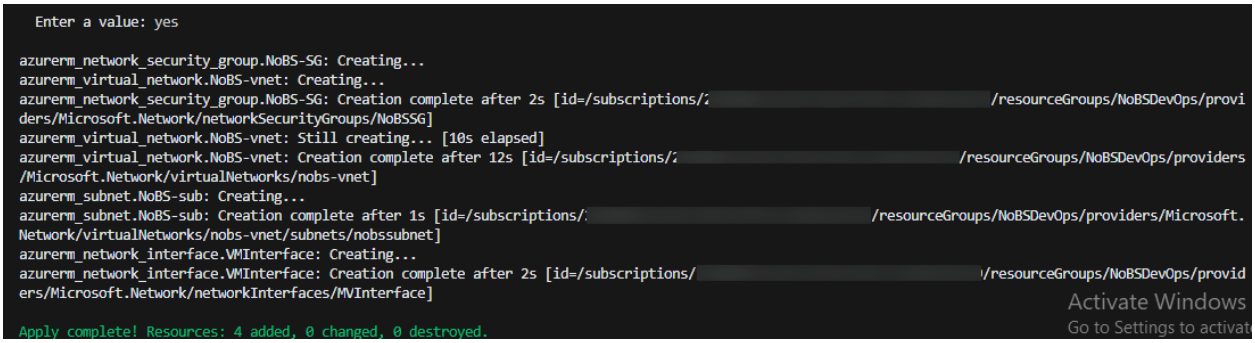
1. Open up a command prompt, a PowerShell console or use the VS Code integrated terminal.
2. Ensure the current working directory is *vnet*. Once inside of the *vnet* directory, run `terraform init` to initialize the Terraform environment and to download the `azurerm` provider.



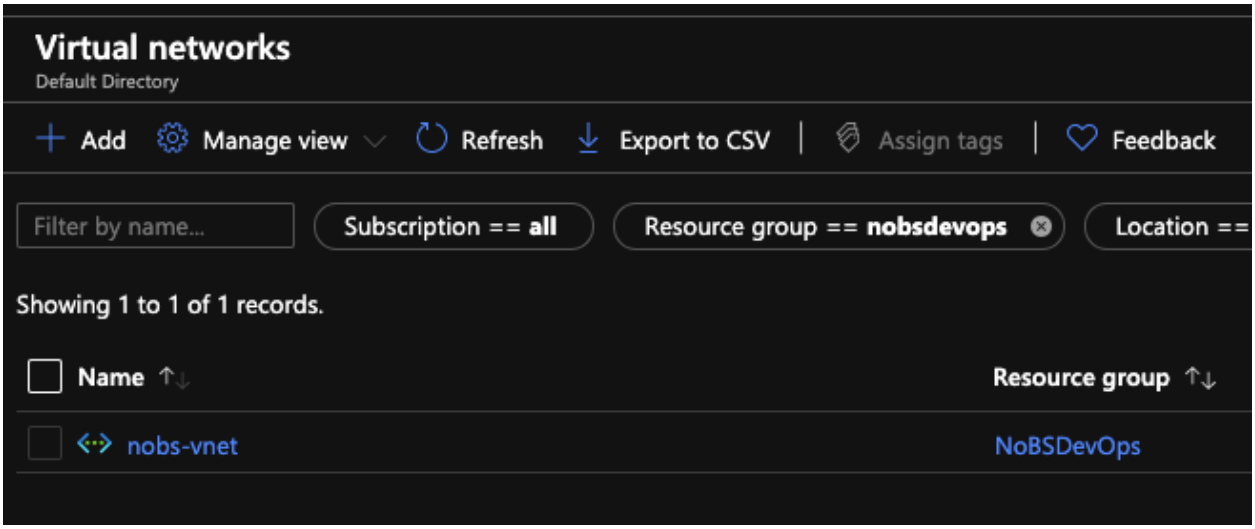
3. Next, run `terraform plan` to validate the configuration and to see what resources will be created. The output should be similar to the screenshot below. This shoud all show nice, green plus symbols.

```
  # azurerm_subnet.NoBS-sub will be created
  + resource "azurerm_subnet" "NoBS-sub" {
      + address_prefix                                 = "10.0.1.0/24"
      + enforce_private_link_endpoint_network_policies = false
      + enforce_private_link_service_network_policies  = false
      + id                                             = (known after apply)
      + name                                           = "nobssubnet"
      + resource_group_name                            = "NoBSDevOps"
      + virtual_network_name                           = "nobs-vnet"
    }

  # azurerm_virtual_network.NoBS-vnet will be created
  + resource "azurerm_virtual_network" "NoBS-vnet" {
      + address_space       = [
          + "10.0.0.0/16",
        ]
      + dns_servers         = [
          + "8.8.8.8",
          + "8.8.4.4",
        ]
      + id                  = (known after apply)
      + location            = "eastus"
      + name                = "nobs-vnet"
      + resource_group_name = "NoBSDevOps"
      + tags                = {
          + "environment" = "Dev"
        }

      + subnet {
          + address_prefix = (known after apply)
          + id             = (known after apply)
          + name           = (known after apply)
          + security_group = (known after apply)
        }
    }

Plan: 5 to add, 0 to change, 0 to destroy.
```

4. Run `terraform apply`, type "yes", and hit Enter. Terraform will then reach out to Azure and begin creating the vNet. Once the resource is created, you should see a screenshot similar to the one below.



5. Go to the Azure portal and to the Virtual Networks Azure blade[32]. You should now see a newly-created vNet, similar to the screenshot below.



Resources

vNet Resource with Terraform[33]

Getting Started with Terraform[34]

[32] https://portal.azure.com/#blade/HubsExtension/BrowseResource/resourceType/Microsoft.Network%2FvirtualNetworks
[33] https://www.terraform.io/docs/providers/azure/r/virtual_network.html
[34] https://www.terraform.io/intro/index.html

Ansible

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

What Makes Ansible Different

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Ansible Can Reach Many Targets

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Ansible Authentication and It's Agentless Approach

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Playbooks

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Ansible Syntax

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Inventory

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Modules

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Variables

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Creating an Azure Container Instance (ACI)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Project Overview

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Tools to Have

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Writing the Playbook

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Running the Playbook

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Verifying Playbook Results

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Resources

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Azure Load Balancers

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

The Basics of Azure Load Balancer

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Creating Duplicate Copies of a Web Service

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Deciding What Instance to Route Traffic To

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Accepting Requests from Clients

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Distributing Traffic to the Backend Pool

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Handling Specific Inbound Port Mappings

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Handling Specific Outbound Port Mappings

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Project Overview

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Tools to Have

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Preparing to Build the Azure Resources

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Resource Group

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Public IP Address

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Networking

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

vNet

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Subnet

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

NSG and Rules

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Azure Load Balancer

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Health Probe

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Load Balancer Rules

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

VM Availability Set

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Testing the Load Balancer

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Cleaning Up

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Project Conclusion

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Resources

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Monitoring in Azure

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Built-In Azure Resource Monitoring

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Azure Monitor

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Azure Monitoring is Cheap

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Subscription Monitoring

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Azure Monitor for VMs

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Application Insights

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Monitoring Containers

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Monitoring with Azure Kubernetes Service (AKS)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Bringing All Monitoring Together

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Setting up Monitoring for a Containerized Web Application

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Tools to Have

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Creating an Azure Container Instance

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Viewing Container Monitoring Metrics

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Resources

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Azure Pipelines

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Builds

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Testing

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Continuous Integration (CI)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Continuous Delivery/Deployment

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Pipelines in Azure

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

YAML Pipelines

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Creating a Pipeline to Deploy an Azure Virtual Machine (VM)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Project Overview

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Tools to Have

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Building an Azure DevOps Organization

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Building the CI/CD pipeline with YAML

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Creating Variables

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Setting Up Pipeline Tasks

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Deploying the Pipeline

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Docker

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

The Docker Engine

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Docker Images

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

DockerFile

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

What Docker Isn't

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Deploying A .NET Core Container

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Project Overview

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Tools to Have

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Installing Docker

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Building the Dockerfile

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Creating a Docker Image

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Running a Docker Container

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Conclusion

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Resources

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Azure Container Registry

Docker Registries and Versioning

Azure Container Registry

Project Overview

Prerequisites

Creating a Resource Group

Creating an ACR

Logging into ACR

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Tagging and Pushing a Docker Image

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Inspecting the New ACR Image

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Cleaning Up

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Kubernetes

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Understanding Kubernetes

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Nodes

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Master Nodes

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Worker Nodes

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Manifests

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

An Example Manifest

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Etcd

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Pods

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Deployments

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Services

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Managed Solutions

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Testing Deployments with Minikube

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Project Overview

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Tools to Have

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Working with the Kubernetes Command Line (Kubectl)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Downloading Kubectl

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Kubctl Command Syntax

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Enabling Hyper-V

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Installing Minikube

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Running Minikube

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Testing Minikube

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Deploying a Docker Image into Kubernetes

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Resources

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Part II - Introduction

If you've made it this far in the book, congratulations! By now you should have a solid, foundational knowledge of many of the most popular services and tools you can use to deliver value to customers with DevOps.

But that was the easy part…

You've made it to Part II. Part II is where the rubber meets the road. We will now see what DevOps is like in the real world.

In this Part, you're going to be thrown into a common situation many new DevOps pros find themselves in. You'll be on a team that has inherited an existing web application without a CI/CD pipeline in sight or hardly any automation at all. In fact, all developers on this web app running on virtual machines don't do much testing at all. They just right-click and publish in Visual Studio!

Building the Monolith

To demonstrate how to "run some DevOps" on your application deployment process and to maintain a build and release pipeline the modern way, we first need a foundation to build upon.

Since it's unlikely you're going to be starting from scratch in the real world, we need a legacy application built like a monolith to migrate infrastructure to. Building this monolith is what the first chapter of Part II is going to be all about.

The fictional application we're assuming you've inherited in the real world is called MovieApp[35]. MovieApp is a .NET Core application a development team has created to provide a Netflix-like interface to various movies your fictional company releases. This application is public-facing and allows paying customers to log in and browse your company's movie selections.

Your organization currently doesn't have a testing environment for this application and deploy straight to production whenever they feel like it.

Building the New Standard

In the second chapter of Part II is where the real fun begins! In this chapter is where you're going to learn all of the nitty-gritty details of building a brand new, modern infrastructure to support the MovieApp application.

You'll be building:

[35] https://github.com/NoBSDevOps/movieapp-dotnet-core

- A Docker image
- An Azure Kubernetes cluster
- A build/release pipeline to deploy MovieApp to Azure

The *Building the New Standard* chapter will be the culmination of everything you've learned in Part I and how to apply all of that knowledge to a real project!

Building the Monolith

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

MovieApp Architecture

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Infrastructure

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Application

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Building the Monolith

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Prerequisites

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Building the Infrastructure

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Downloading the Terraform Configuration

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Building the Infrastructure with Terraform

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Installing and Configuring IIS

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Deploying the Application

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Wrap Up

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Building the New Standard

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Prerequisites

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Project Overview

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Setting up the Docker Image

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Building the Docker Image

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Adding the Docker Image to Azure Container Registry (ACR)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Creating an Azure KeyVault with Secrets

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Build the KeyVault

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Create the Service Principal for AKS

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Adding Secrets to the KeyVault

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Building an Azure Kubernetes Cluster via Pipeline

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Building the Terraform Plan

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Building the Azure DevOps Pipeline

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Building the Pipeline for Deploying MovieApp

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Verifying the MovieApp

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Monitoring the MovieApp

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Wrap Up

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/fromadmintodevops.

Release Notes

02/25/21
- Added the final New Standard project

12/09/20

- Added Part II intro

11/30/20

- Added the Part I project

10/03/20
- Added Azure Container Registry chapter

06/05/20
- Added Azure Monitoring chapter

05/21/20
- Added Azure Pipelines chapter

05/15/20
- Added Ansible chapter

05/08/20
- Added Azure VM Availability Sets chapter

04/27/20
- Added Terraform chapter

04/23/20
- Added Visual Studio Code chapter

04/14/20
- Added Kubernetes chapter

04/09/20
- Added Part 1: Tools

04/07/20
- Added the Azure Load Balancers chapter

04/01/20
- Aded Git and GitHub chapter

03/29/20
- Scaffolded out Part II milestones to provide a glimpse into where it's heading

03/28/20
- Added Docker chapter

03/27/20
- Added the Azure from the commamnd line with cloud shell chapter

03/24/20
- sample book creation and formatting fixes

03/20/20
- added the first few chapters and all of the intro sections