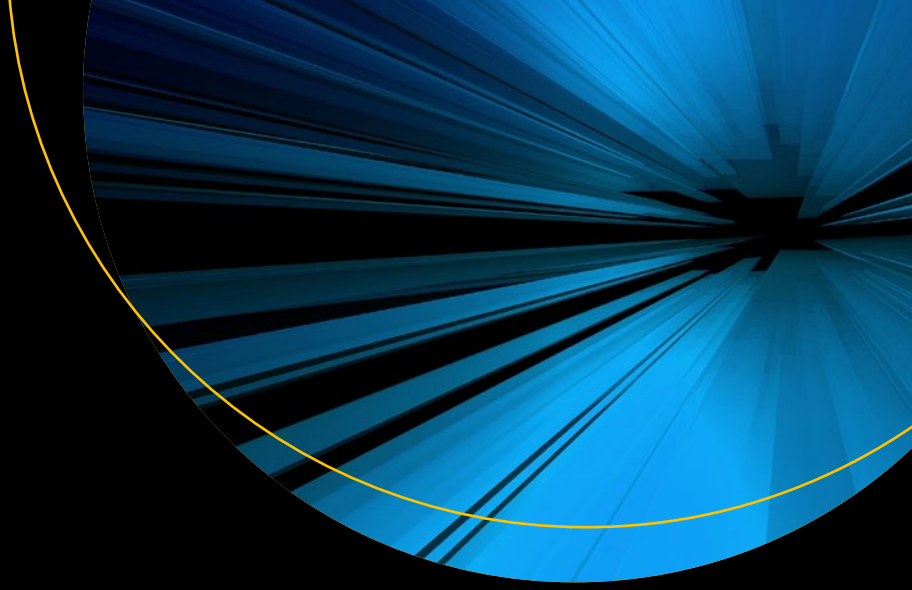


Migrating a Two-Tier Application to Azure

A Hands-on Walkthrough of Azure Infrastructure, Platform, and Container Services

—

Peter De Tender

Apress®

Migrating a Two-Tier Application to Azure

A Hands-on Walkthrough of Azure Infrastructure, Platform, and Container Services

Peter De Tender

Apress®

Migrating a Two-Tier Application to Azure

Peter De Tender
Lokeren, Belgium

ISBN-13 (pbk): 978-1-4842-6436-2 ISBN-13 (electronic): 978-1-4842-6437-9
https://doi.org/10.1007/978-1-4842-6437-9

Managing Director, Apress Media LLC: Welmoed Spahr
Acquisitions Editor: Spandana Chatterjee
Development Editor: Matthew Moodie
Coordinating Editor: Divya Modi

Cover designed by eStudioCalamar

Cover image designed by Pixabay

Distributed to the book trade worldwide by Springer Science+Business Media New York, 1 New York Plaza, New York, NY 10004. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com. Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a **Delaware** corporation.

For information on translations, please e-mail booktranslations@springernature.com; for reprint, paperback, or audio rights, please e-mail bookpermissions@springernature.com.

Apress titles may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Print and eBook Bulk Sales web page at http://www.apress.com/bulk-sales.

Any source code or other supplementary material referenced by the author in this book is available to readers on GitHub via the book's product page, located at www.apress.com/978-1-4842-6436-2. For more detailed information, please visit http://www.apress.com/source-code.

Printed on acid-free paper

I dedicate this book to Ivan De Rop, head teacher in my senior year in high school, for believing in my skills and passion for information technology, although I did business management studies.

Table of Contents

About the Author

Peter De Tender has more than 20 years of experience in architecting and deploying Microsoft datacenter technologies. Since early 2012, he started shifting to cloud technologies (Office 365, Intune) and quickly jumped onto the Azure platform, working as cloud solution architect and trainer, out of his own company. Since September 2019, Peter moved into an FTE role within Microsoft Corp in the prestigious Azure Technical Trainer team, providing Azure readiness workshops to larger customers and partners across the globe.

Peter was an Azure MVP for 5 years and IS a Microsoft Certified Trainer for more than 12 years and is still actively involved in the community as speaker, technical writer, and author.

You can follow Peter on Twitter @pdtit and check his technical blog, https://www.007FFFLearning.com.

About the Technical Reviewer

Amita Thukral is an IT professional, an NIIT degree holder, and ITIL certified. She has more than 16 years of extensive experience working with top IT organizations like Wipro Infotech, Dell India, Hughes Software Systems, and Xcad Agencies. She worked as a technical editor for Leanpub Publishing with author Peter De Tender (MVP) for a web book "Migrating a dotnetcore 2-tier application to Azure, using different architectures and DevOps best practices."

As a service delivery manager, she has handled multiple IT instructor–led and online trainings across various global locations. As a project manager, she was responsible for running cloud computing projects, like Azure and Dynamics 365, and prepared comprehensive action plans, including resources, timeframes, and budgets for projects. She worked on updating, reviewing, and building documentation and content of the lab guides and ebooks for several cloud-based technical projects. She has performed coordinating tasks like planning and scheduling, along with administrative duties like maintaining project documentation, database management, and collaborating with clients and internal teams to deliver results. She ensured that all projects were completed on time and within budget and met high-quality standards.

Acknowledgments

After writing seven technical books, it's hard to come up with original thank-you words. Anyone reading this book knows this is a work of time, dedication, and passion for technology, as well as a passion for sharing knowledge. I am fortunate enough to have a wife supporting me in this. But I'm no longer allowed to thank her (her own words), as sharing knowledge and helping people is what makes me who I am.

That said, I owe a big thanks to Spandana Chatterjee and Divya Modi from Apress, who picked up my "Azure hands-on labs" self-published material from Leanpub and offered to take over the content and publish it through Apress. This was the best opportunity to update the technical content, make it current, and add new topics to the exercises. And your audience reach-out is much broader than what I could ever get myself, so you help in spreading the Azure knowledge.

I'd also like to thank my technical reviewer, Amita Thukral – my faraway friend from India, always eager to help where she can and at the same time living the "continuous learning" life. You are professional, have an amazing drive for details, and are overall a lovely person to work with.

And Wim Matthyssen, community buddy and fellow Azure expert, thanks for jumping in last minute to give your technical blessing on the flow, wording, and lab scenarios and overall validate them.

Both of you pushed up the level of quality.

CHAPTER 1

Introduction

Migrating a Two-Tier Application to Azure Using Different Architectures and DevOps Best Practices

Setting the Scene

You are part of an organization that is running an e-commerce platform application, at present using Windows Server on-premises infrastructure, based on a virtual Windows Server 2012 R2 web server running Internet Information Services (IIS) and a second Windows Server 2012 R2 virtual machine (VM) running Microsoft SQL Server 2014 database services.

The business has approved a migration of this business-critical workload to Azure, and you are nominated as the cloud solution engineer for this project. No decision has been made yet on what the final architecture should or will look like. Your first task is building different Proof of Concepts in your Azure environment, to test out the different architectures available today, to host your application workload:

- Infrastructure as a Service (IAAS), using Azure Virtual Machines
- Platform as a Service (PAAS), using Azure Web Apps and Azure SQL
- Containers as a Service (CAAS), using Azure Container Instance (ACI) and Azure Kubernetes Service (AKS)

At the same time, your CIO wants to make use of this project to switch from a more traditional mode of operations, with barriers between IT sysadmin teams and developer teams, to a "DevOps" way of working. Therefore, you are tasked to explore Azure DevOps and determine where CI/CD pipelines, together with other capabilities from Azure DevOps, can assist in optimizing the deployment as well as optimizing the running operations of this e-commerce platform, especially when deploying updates to the application.

P. De Tender, *Migrating a Two-Tier Application to Azure*, https://doi.org/10.1007/978-1-4842-6437-9_1

As you are new to the continuous changes in Azure, you want to make sure this process goes as smooth as possible, starting from the assessment over migration to performing day-to-day operations.

Abstract and Learning Objectives

This book enables anyone to learn, understand, and build a Proof of Concept, by performing a platform migration of a two-tiered application workload to Azure public cloud, leveraging on different Azure Infrastructure as a Service, Azure Platform as a Service (PAAS), and Azure container offerings like Azure Container Instance (ACI) and Azure Kubernetes Service (AKS).

The focus of the book is having a true hands-on lab experience, by going through the following exercises and tasks:

- Deploying your "lab virtual machine"
- Deploying a two-tier Azure Virtual Machine (web server and SQL database server) using Infrastructure as Code (IAC) concepts with ARM (Azure Resource Manager) template automation in Visual Studio 2019
- Performing a proper assessment of the as-is WebVM and SQLVM infrastructure using Microsoft assessment tools
- Migrating a SQL Server 2014 database to Azure SQL PaaS (lift and shift)
- Migrating a .NET Core web application to Azure Web Apps (lift and shift)
- Containerizing a .NET Core web application using Docker and pushing to Azure Container Registry (ACR)
- Running a containerized application in Azure Container Instance (ACI) and Azure Web App for Containers
- Running a containerized application in Azure Kubernetes Service (AKS)

- Deploying Azure DevOps and building a CI/CD pipeline for the sample e-commerce application
- Managing and monitoring Azure Kubernetes Service (AKS) and other Azure Monitor capabilities

Starting from an (optional but highly recommended for consistency) ARM template–based deployment of a lab virtual machine, readers get introduced to the basics of automating Azure resource deployments using Visual Studio and Azure Resource Manager (ARM) templates, together with additional Infrastructure as Code concepts like Custom Script Extension and PowerShell Desired State Configuration (DSC).

Next, readers learn about the importance of performing proper assessments and what tools Microsoft offers to help in this migration preparation phase. Once the application has been deployed on Azure Virtual Machines, readers learn about Microsoft SQL Server database migration to Azure SQL PAAS, as well as deploying and migrating web applications to Azure Web Apps.

After these foundational platform components, the following chapters will totally focus on the core concepts and advantages of using containers for running business workloads, based on Docker, Azure Container Registry (ACR), Azure Container Instance (ACI), and Web App for Containers, as well as how to enable container orchestration and cloud scale using Azure Kubernetes Service (AKS).

In the last part of the book, readers get introduced to Azure DevOps, the Microsoft application lifecycle environment, helping in building a CI/CD pipeline to publish workloads using the DevOps principles and concepts, showing the integration with the rest of the already-touched-on Azure services like Azure Web Apps and Azure Kubernetes Service (AKS), closing the exercises with a chapter on Azure monitoring and operations and what tools Azure has available to assist your IT teams in this challenge.

Note The Proof of Concept lab scenario is built in such a way that each lab exercise is building on top of the previous lab exercise in sequence. Given the dependencies across different labs, make sure you finish each lab exercise successfully, before continuing on to the next lab.

Technical Requirements

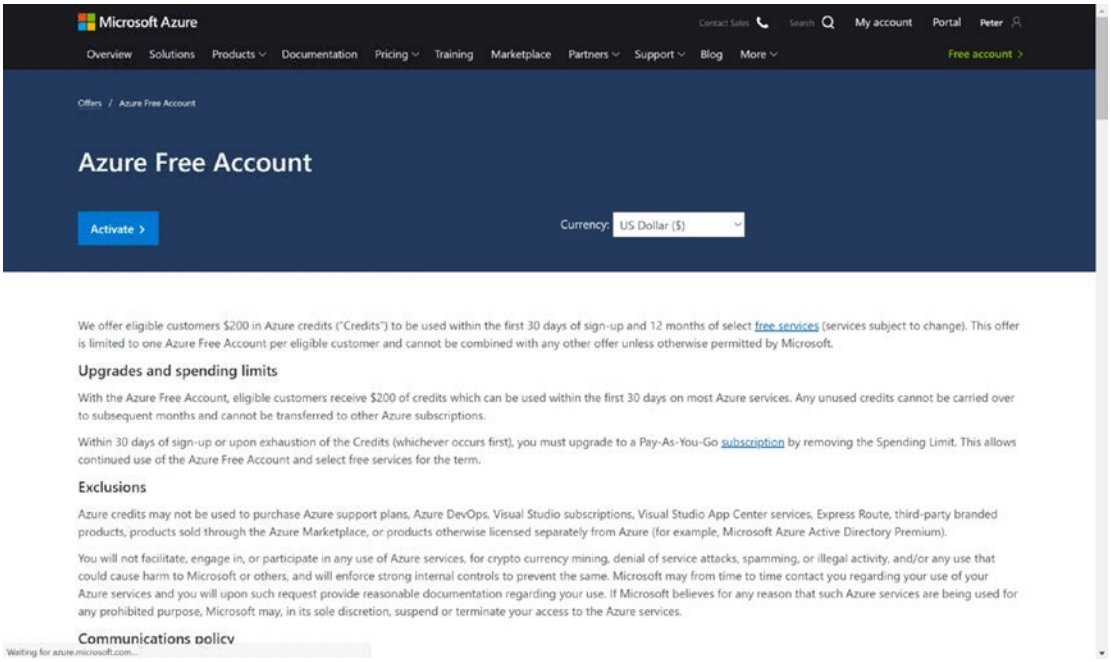
Before being able to perform the hands-on tasks in this book, make sure you meet each of the technical requirements:

- Azure subscription with full administrative permissions
- Naming conventions

Azure Subscription

Make sure you have (full administrative) access to an Azure subscription, allowing you to deploy the different Azure resources being used throughout the exercises. You can use an Azure free or trial subscription or use any paid subscription.

Signing up for a free/trial subscription can be done from here: https://signup.azure.com/signup?offer=ms-azr-0044p&appId=102&l=en-gb&correlationId=37037FE60CF76B40251371B40DDF6AB9



If you go through all exercises, estimate an average consumption of 20–30 USD, assuming you shut down or delete the resources that are no longer in use or required.

Naming Conventions

Important Most Azure resources require unique names. Throughout the lab steps, we will identify the naming convention for the given resources as **"[SUFFIX]"** as part of resource names. You should replace this with a unique string, e.g., your own initials, guaranteeing those resources get uniquely named and not blocking a successful deployment.

Other Requirements

Readers need a local client admin machine, running a recent Operating System, allowing them to

- Browse to `https://portal.azure.com` from a recent browser.
- Establish a secured Remote Desktop (RDP) session to a lab jumpVM running Windows Server 2019.

Alternative Approach

Where the lab scenario assumes all exercises will be performed from within a lab jumpVM (see Chapter 2 on how to get started with this deployment), readers could also execute (most, if not all) steps from their local client machine, if that is what they prefer.

The following tools are being used throughout the lab exercises:

- Visual Studio 2019 community edition (updated to latest version)
- Docker for Windows (updated to latest version)
- Azure CLI 2.0 (updated to latest version)
- Kubernetes CLI (updated to latest version)
- SimplCommerce Open Source e-commerce platform example (`http://www.simplcommerce.com`)

Note Make sure you have these tools installed prior to the workshop if you are not using the lab jumpVM. You should also have full administrator rights on your machine to execute certain steps in using these tools.

Final Remarks

Due to the continuously evolving nature of Azure, Azure services, the Azure Portal, and other tools we will be using for the exercises, it might be that some screenshots or wordings do not match what you will see on your end. We apologize for this already, although there isn't much we can do about it. If the differences are too many, it would be almost impossible to execute the exercises. Please have a look at our GitHub repository `http://www.apress.com/source-code` for any updates and errata.

We hope you enjoy the different exercises, learn from them, and find them useful in your day-to-day job or journey in which you explore Azure capabilities. Do not hesitate reaching out at peter@pdtit.be or @pdtit (Twitter) in case you have any questions. We are here to help you making this a successful learning path.

CHAPTER 2

Prerequisite Lab: Deploying Your Lab Virtual Machine

Prerequisite lab: Preparing your (Azure) environment

What You Will Learn

In this first lab, you prepare the baseline for executing all hands-on lab exercises:

- Log on to your Azure subscription.
- Deploy the lab jumpVM within your Azure subscription.
- Download the required source files from GitHub to the lab jumpVM.

Time Estimate

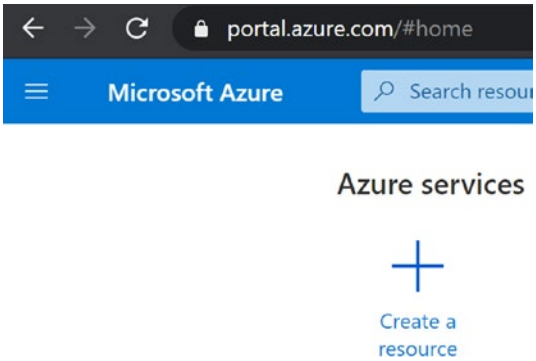
This lab is estimated to take **45 min**, assuming your Azure subscription is already available.

P. De Tender, *Migrating a Two-Tier Application to Azure*, https://doi.org/10.1007/978-1-4842-6437-9_2

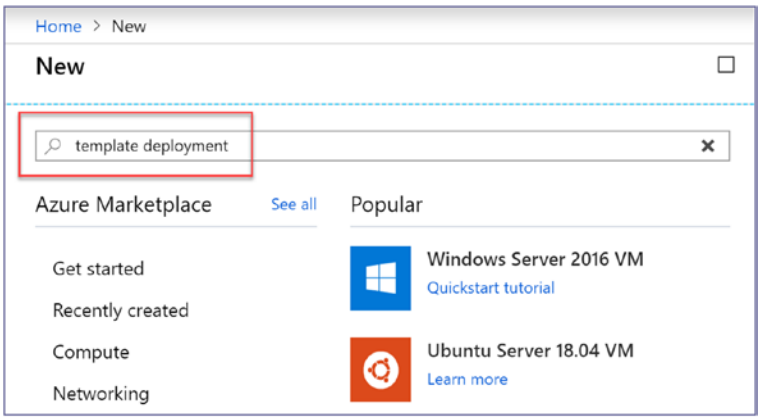
Task 1: Deploying the lab jumpVM virtual machine using Azure Portal template deployment

In this task, you start deploying the "lab jumpVM" virtual machine in your Azure environment. This machine becomes the starting point for all future exercises, as it has most required tools already installed. The deployment is based on an ARM (Azure Resource Manager) template in a publicly shared GitHub repository.

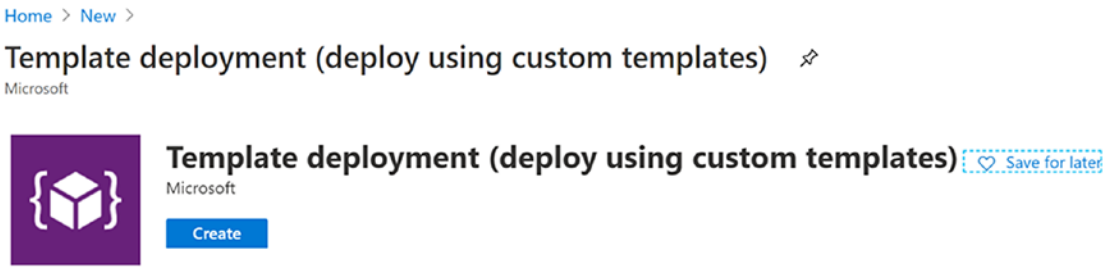
1. Once you are logged on to your Azure subscription, select **Create a Resource**.



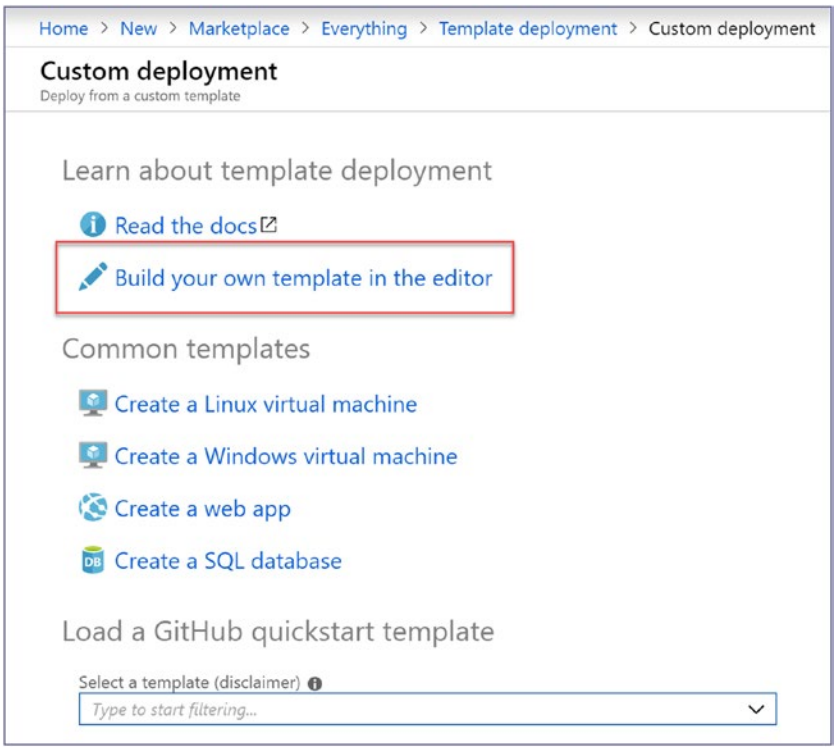
2. In the Search Azure Marketplace field, type "template deployment".



3. And select **Template deployment (deploy using custom templates)** from the list of Marketplace results, followed by clicking the **Create** button.

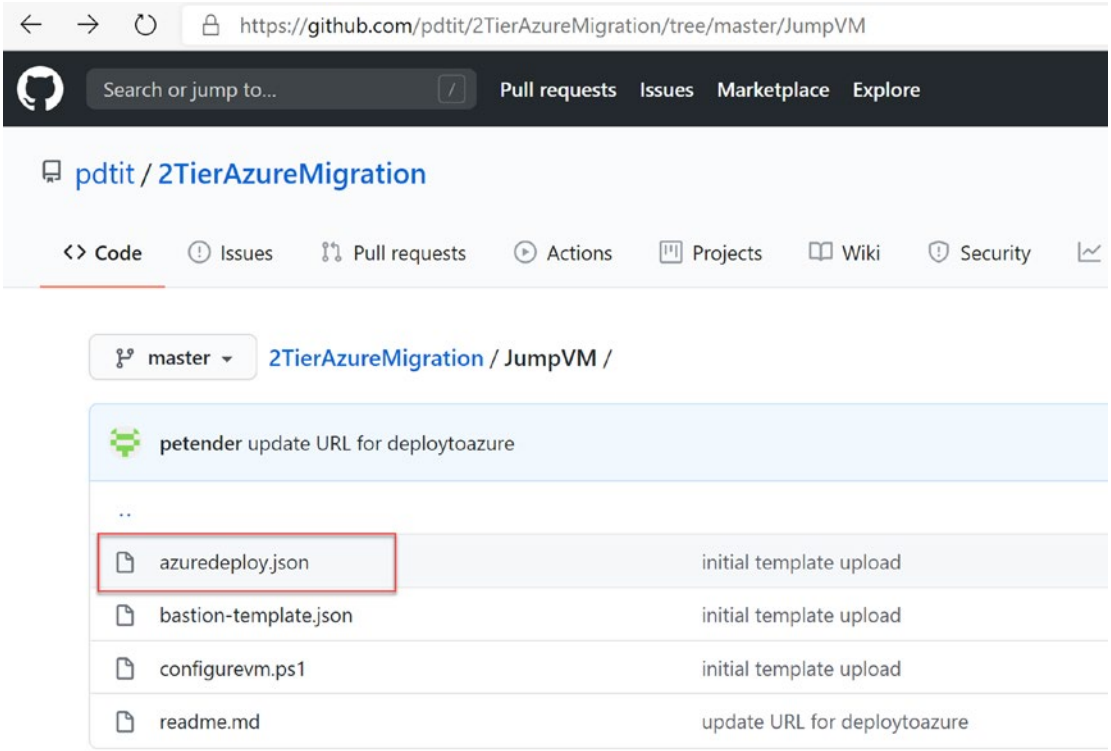


4. This opens the Custom deployment blade. Here, select "Build your own template in the editor."

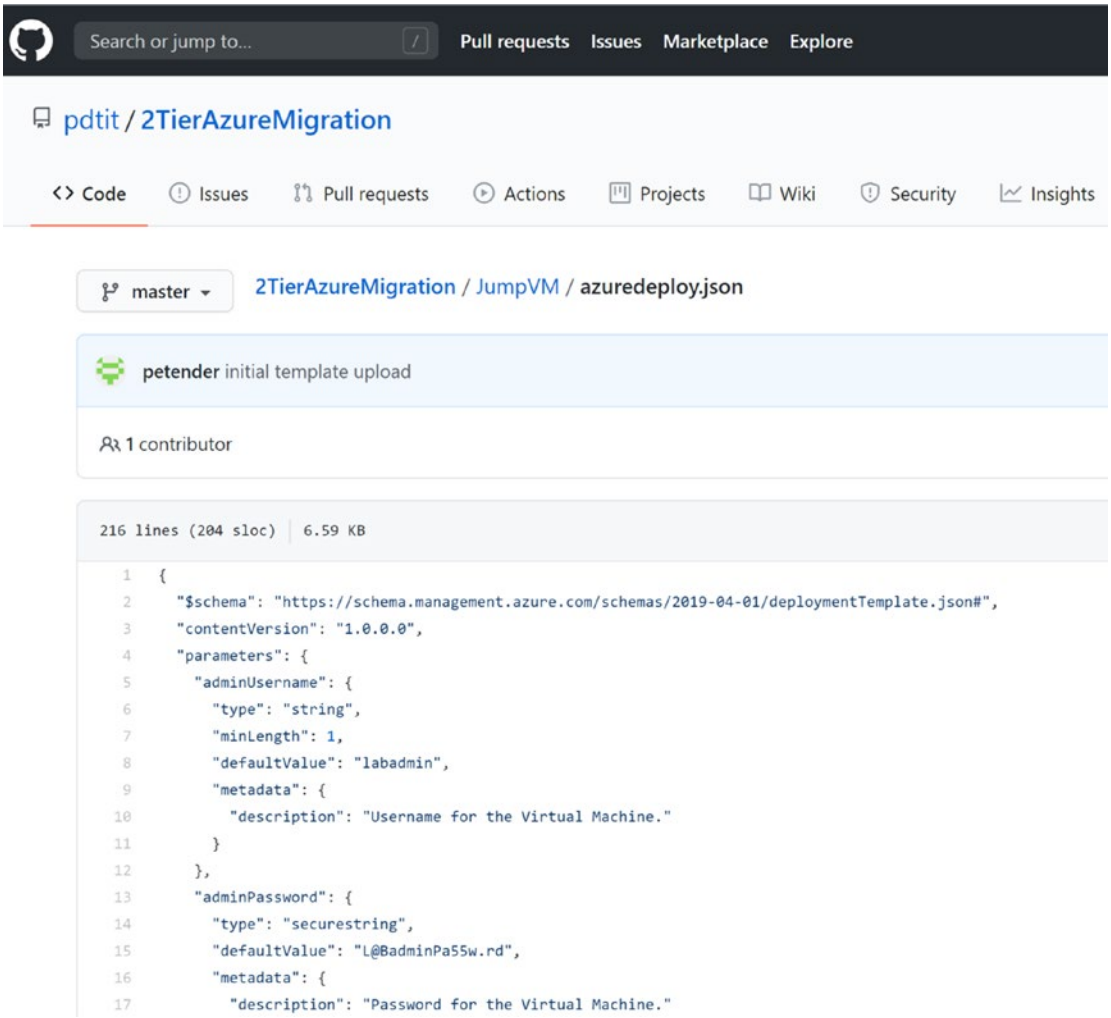


5. First, from a **second tab** in your browser window, go to the following URL on GitHub, browsing to the source files repository for this lab, specifically the JumpVM folder:

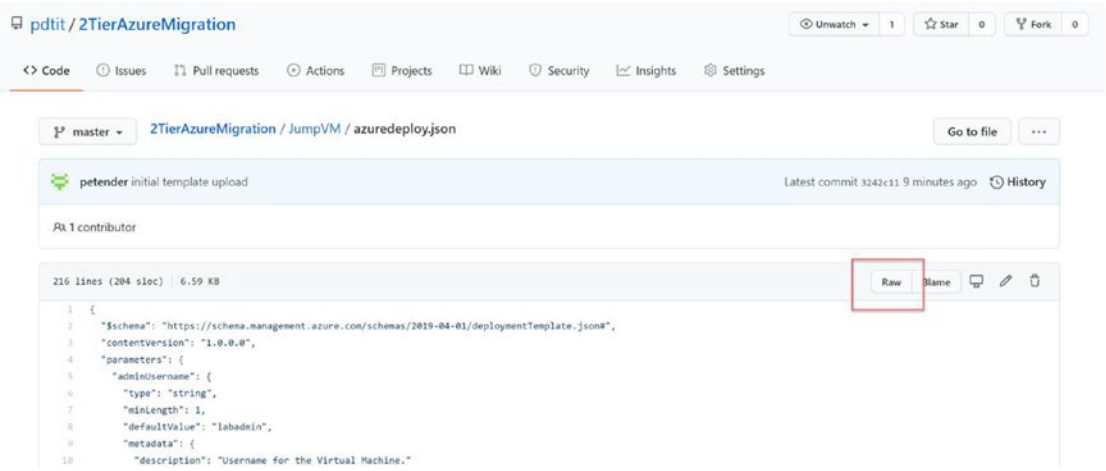
   `http://www.apress.com/source-code`.



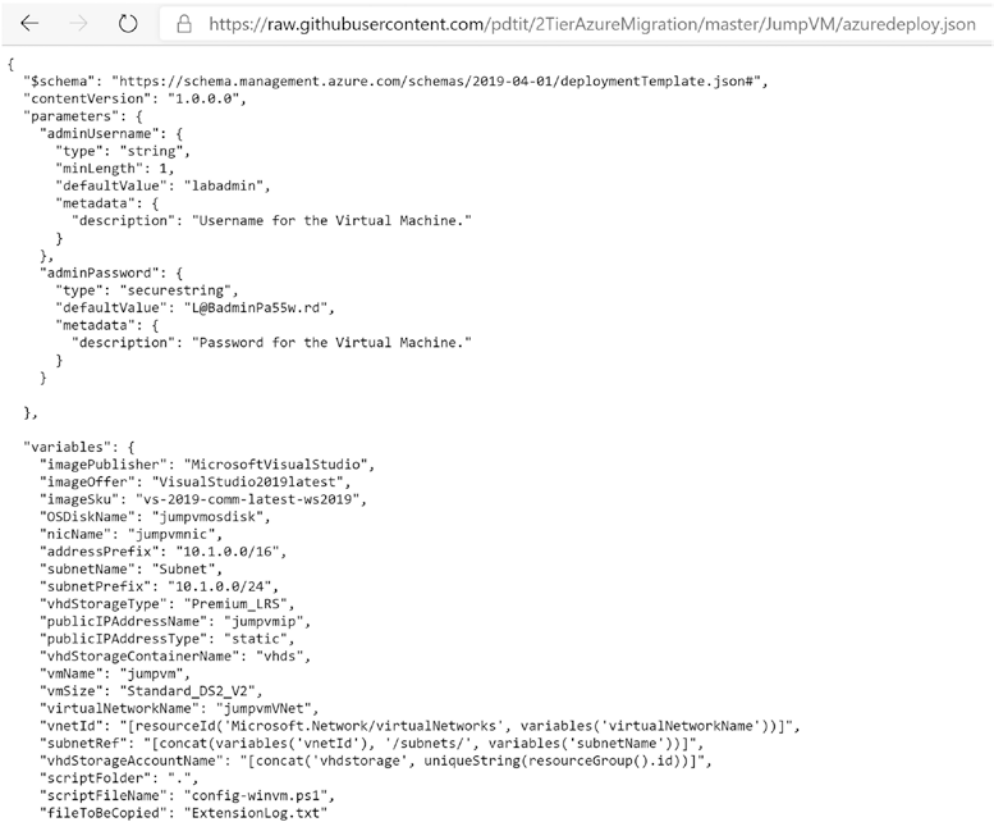
6. **Select** the azuredeploy.json object in there. This exposes the details of the actual JSON deployment file.



7. **Click** the **Raw** button, to open the actual file in your browser.

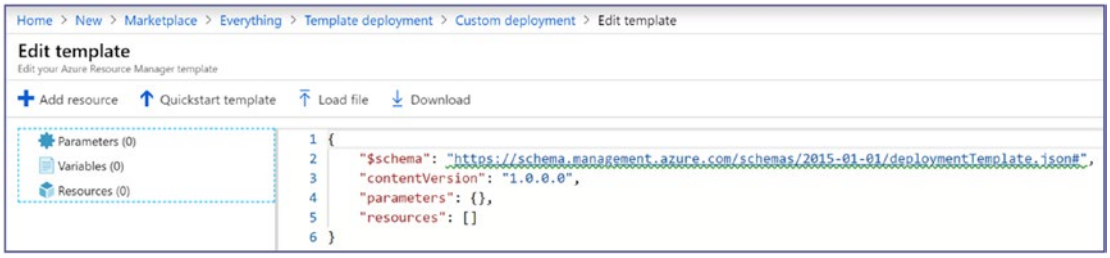


8. Your browser should show the content as follows:

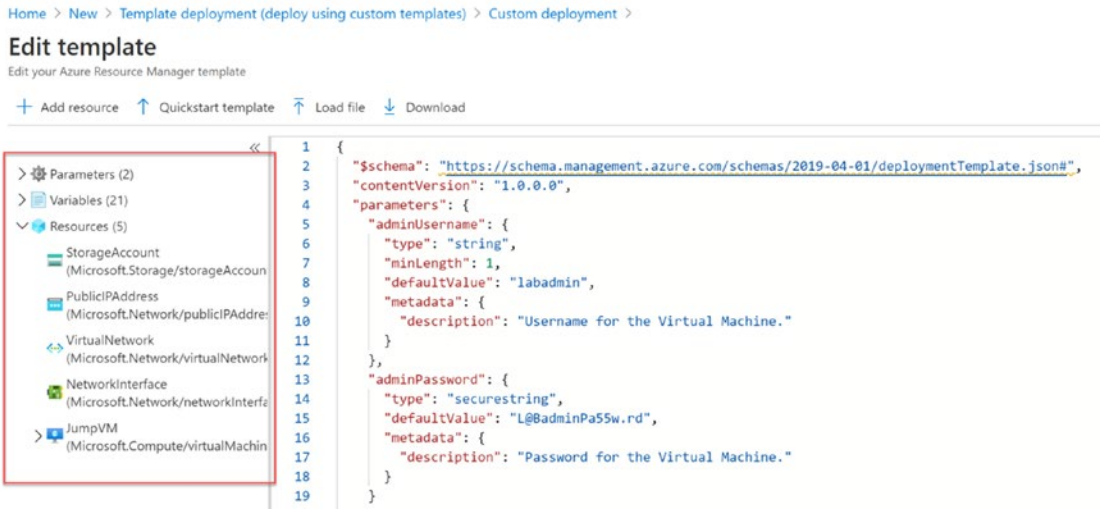


9. Here, **select all lines in the JSON** file, and copy its content to the clipboard.

10. **Go back** to the **Azure Portal**. From "the **edit template**" blade, **remove the first six lines of code** you see in there, and **paste in the JSON content from the clipboard.**

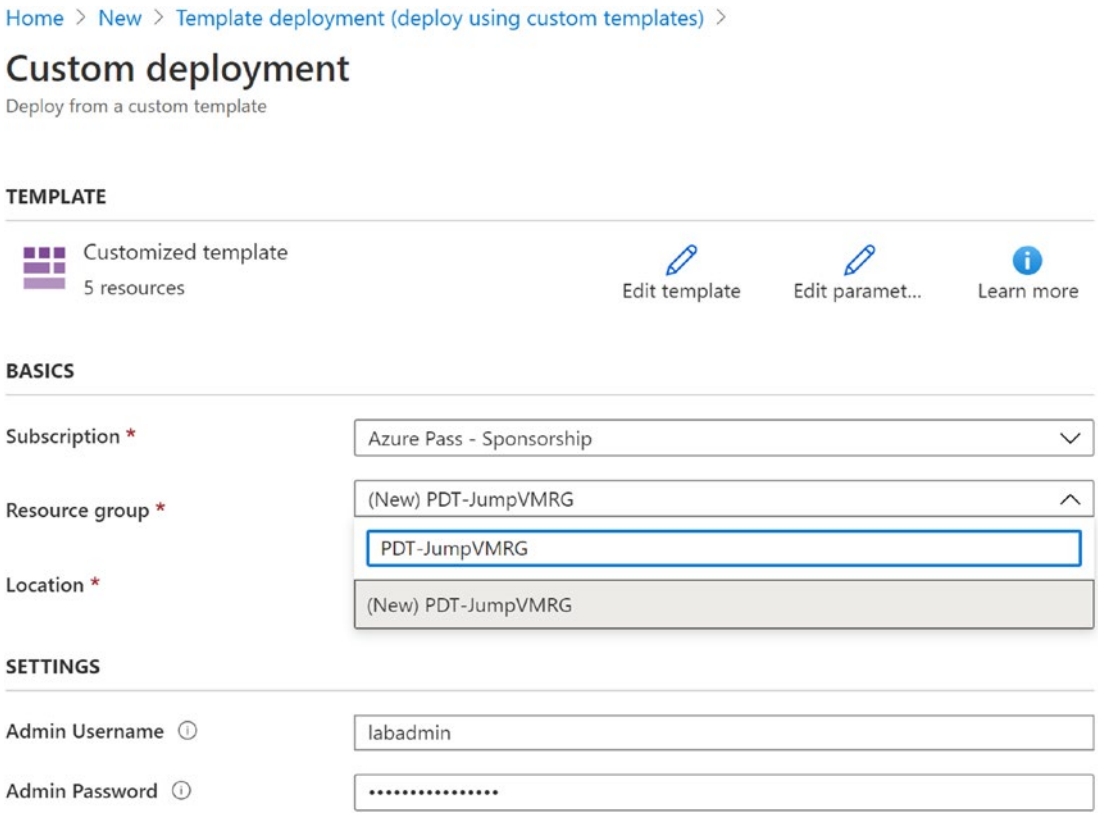


11. "The edit template" blade should recognize the content of the JSON file, showing the details in the JSON Outline on the left.

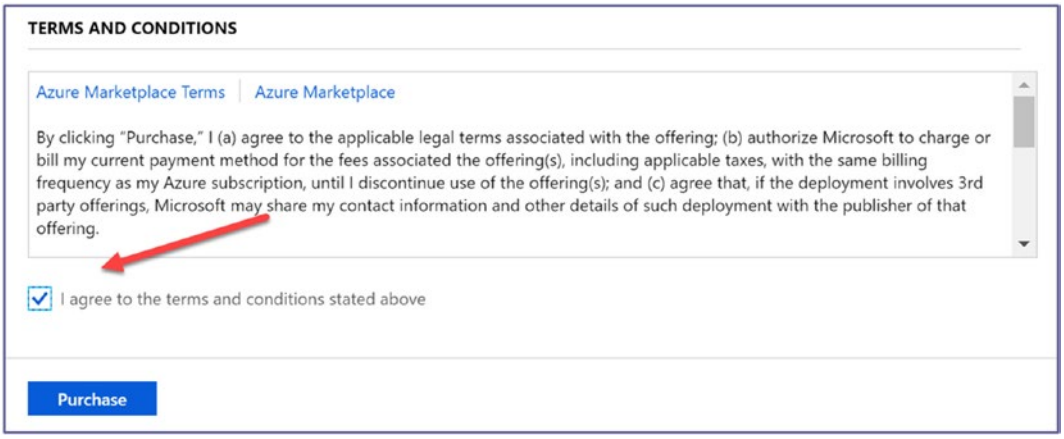


12. **Click** the **Save** button.

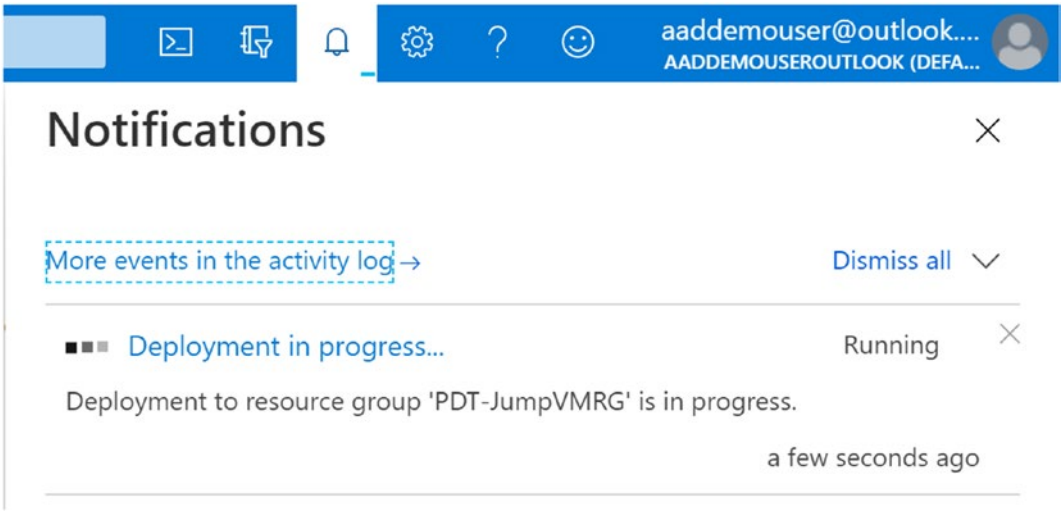
13. This **redirects** you back to the Custom deployment blade, from where you will **execute** the actual template deployment, filling in the required fields as follows:
    - **Subscription: Your Azure subscription**
    - **Resource group: Create New/[SUFFIX]-JumpVMRG**
    - **Location: Your closest by Azure region**
    - **Admin Username: labadmin** (this information is picked up from the ARM template; although you could change this, we recommend you to not do so for consistency with the lab guide instructions and avoiding any errors during later deployment steps)
    - **Admin Password:** L@BadminPa55w.rd (this information is picked up from the ARM template; although you could change this, we recommend you to not do so for consistency with the lab guide instructions and avoiding any errors during later deployment steps)



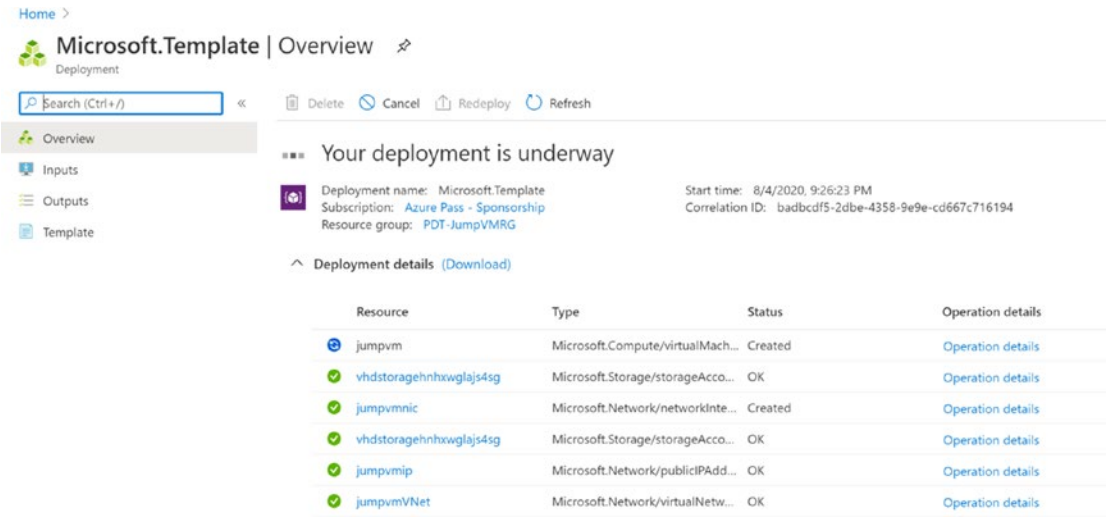
14. When all fields have been completed, scroll down in the blade. Under the Terms and Conditions section, **check "I agree to the terms and conditions stated above,"** and **click** the **Purchase** button.



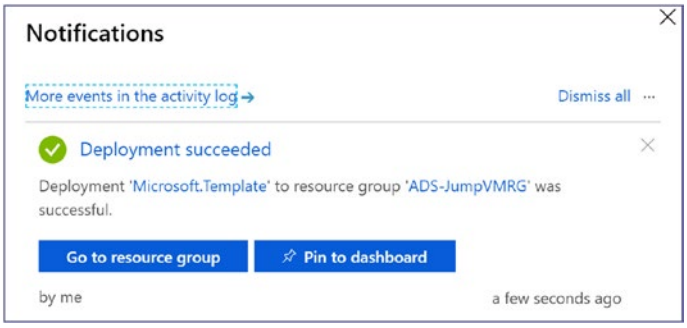
15. This sets off the actual **Azure resource deployment process**. From the **Notifications area**, you can get update information about the deployment.



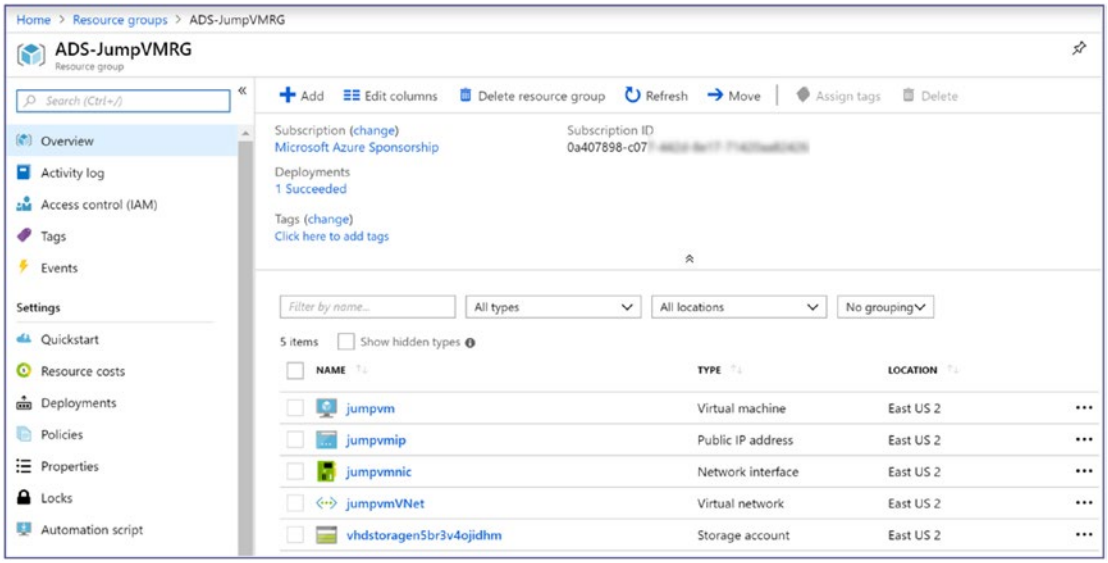
16. If you click "Deployment in progress…," you will get redirected to the Microsoft.Template Overview blade, showing you the details of each Azure resource getting deployed.



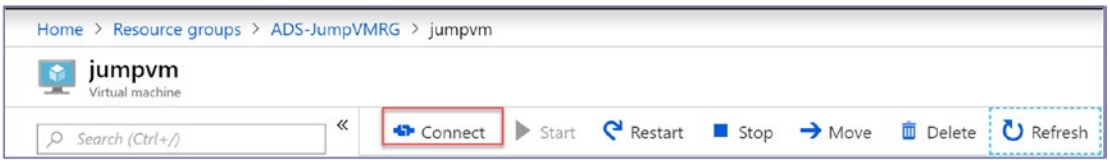
17. Wait for the deployment to complete successfully. **Note this could take up to 25–30 minutes, because of the custom scripts we run during the installation** process, which you can see from this detailed view or from the Notifications area.



18. From the notification message, **click** "Go to resource group." (If you already closed the notification message, from the Azure Portal navigation menu to the left, select Resource groups.)



19. **Click** the **jumpvm Azure Virtual Machine** resource. This redirects you to the detailed blade for the jumpvm resource. Here, **click** the **Connect** button.

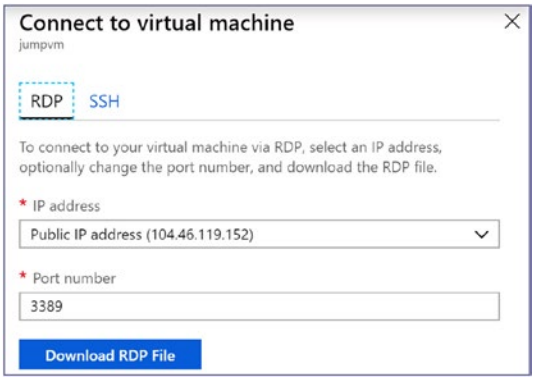


Note Because the VM is linked to a "basic" public IP address resource, all incoming TCPIP traffic is allowed. Therefore, incoming RDP is just working. In a real-life scenario, this VM would be configured with Network Security Group (NSG) rules, only allowing specific traffic.

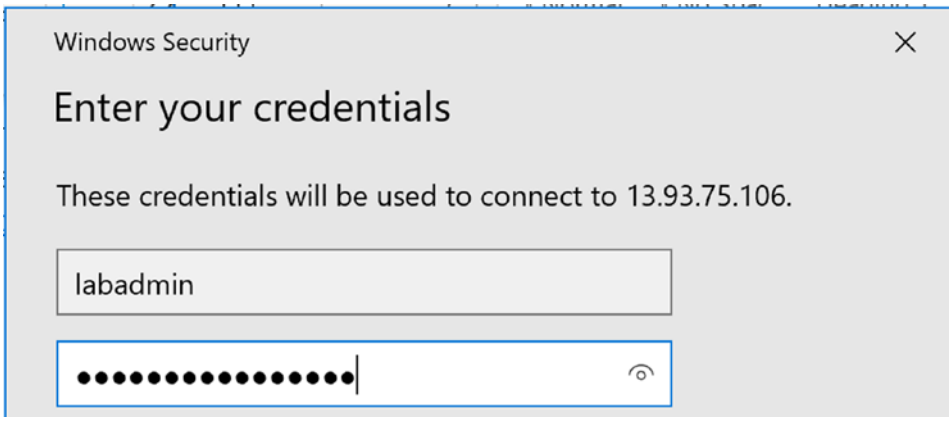
---

20. From the **Connect to virtual machine** blade, notice the **public IP address and port 3389**. This allows you to establish an RDP session to the Azure VM. Do this by **clicking** the **Download RDP File** button.

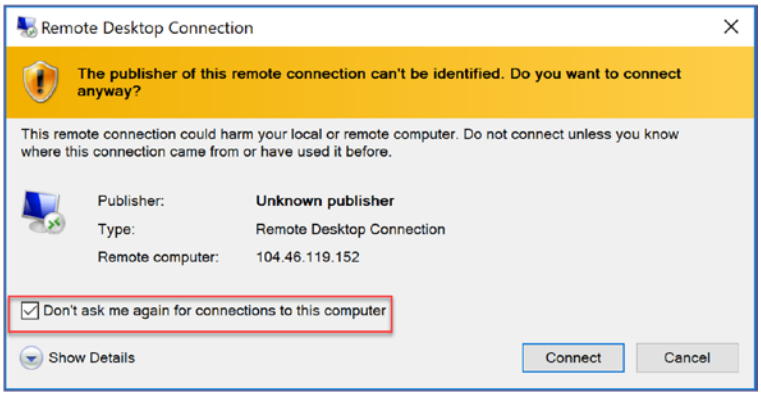
(**Note:** If your local network blocks direct RDP to Azure VMs, consider having a look at Azure Bastion, an Azure service performing HTML5 browser-based routing to RDP or SSH-enabled machines. Specifically for this JumpVM, we offer an ARM template in the same GitHub repo as the JumpVM: `https://github.com/pdtit/2TierAzureMigration/blob/master/JumpVM/bastion-template.json`.



21. **Open** the downloaded RDP file; You are prompted for your credentials in the next step, provide the **VM administrator name (labadmin) and its password (**L@BadminPa55w.rd**), which are the default**.



22. From the appearing popup window, set the flag to "Don't ask me again for connections to this computer."



23. Your Remote Desktop session to this Azure VM gets established.

24. A popup message will appear, asking if you want to allow network discovery; close this popup using the **No** button.