**Implementation of Three-Tier Architecture Using ARM Template**

**Team Members:**

|  |  |  |
| --- | --- | --- |
| **Name** | **SAP ID** | **Branch** |
| Ashwani Singh | 500068763 | CSE CCVT |
| Amaan Ansari | 500067262 | CSE CCVT |
| Gaurav Gupta | 500068658 | CSE CCVT |
| Aysuh Kumar Todariya | 500069522 | CSE BFSI |

**Mentors:**

**Ms. Anushree Sah**

**Mr. Tushar Mittal**



School of Computer Science

**UNIVERSITY OF PETROLEUM AND ENERGY STUDIES**

**Dehradun-24800**

**ABSTRACT**

Three-tier architecture is a well-established software application architecture that organizes applications into three logical and physical computing tiers: the presentation tier, or user interface; the application tier, where data is processed; and the data tier, where the data associated with the application is stored and managed.

The chief benefit of three-tier architecture is that because each tier runs on its own infrastructure, each tier can be developed simultaneously by a separate development team, and can be updated or scaled as needed without impacting the other tiers.

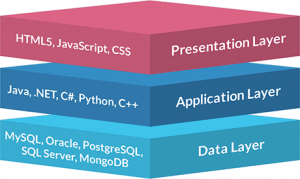
To implement infrastructure as code for your Azure solutions, use Azure Resource Manager templates (ARM templates). The template is a JavaScript Object Notation (JSON) file that defines the infrastructure and configuration for your project. The template uses declarative syntax, which lets you state what you intend to deploy without having to write the sequence of programming commands to create it. In the template, you specify the resources to deploy and the properties for those resources.

**INTRODUCTION**

A 3-tier architecture is a type of software architecture which is composed of three “tiers” or “layers” of logical computing. They are often used in applications as a specific type of client-server system. 3-tier architectures provide many benefits for production and development environments by modularizing the user interface, business logic, and data storage layers. Doing so gives greater flexibility to development teams by allowing them to update a specific part of an application independently of the other parts. This added flexibility can improve overall time-to-market and decrease development cycle times by giving development teams the ability to replace or upgrade independent tiers without affecting the other parts of the system.

**T****hree Tier Architecture Components:**

* **Presentation Tier-**The presentation tier is the front end layer in the 3-tier system and consists of the user interface. This user interface is often a graphical one accessible through a web browser or web-based application and which displays content and information useful to an end user. This tier is often built on web technologies such as HTML5, JavaScript, CSS, or through other popular web development frameworks, and communicates with others layers through API calls.
* **Application Tier-**The application tier contains the functional business logic which drives an application’s core capabilities. It’s often written in Java, .NET, C#, Python, C++, etc.
* **Data Tier-**The data tier comprises of the database/data storage system and data access layer. Examples of such systems are MySQL, Oracle, PostgreSQL, Microsoft SQL Server, MongoDB, etc. Data is accessed by the application layer via API calls.



**Microsoft Azure**

Microsoft Azure, commonly referred to as Azure also is a cloud computing service created by Microsoft for building, testing, deploying, and managing applications and services through Microsoft-managed data centers. It provides software as a service (SaaS), platform as a service (PaaS) and infrastructure as a service (IaaS) and supports many different programming languages, tools, and frameworks, including both Microsoft-specific and third-party software and systems.

Azure, announced at Microsoft's Professional Developers Conference (PDC) in October 2008, went by the internal project codename "Project Red Dog", and formally released in February 2010, as Windows Azure before being renamed to Microsoft Azure on March 25, 2014.

* Azure Active Directory is used to synchronize on-premises directories and enable single sign-on.
* Azure Active Directory B2C allows the use of consumer identity and access management in the cloud.
* Azure Active Directory Domain Services is used to join Azure virtual machines to a domain without domain controllers.
* Azure information protection can be used to protect sensitive information.

**ARM Templates**

An ARM template is a JSON (JavaScript Object Notation) script. This script includes the resource(s) that are being deployed to an Azure Resource Group.

In most cases, resources are provisioned from the Azure Portal. But, what if we want to repeat a deployment more than once? At that point the concept of ARM Template came up, which we can use to make the deployments by running a simple PowerShell script.

**How can we Create an ARM Template?**

* There are more than one ways to get started with ARM Template: AZURE Portal (From the Automation Script & Custom deployment).
* GitHub (From the section Azure Quick Templates).
* Visual Studio.
* Visual Studio Code.

**Azure Portal:**

The first place we can view an ARM Template is the Azure Portal. There are several ways to work with ARM Templates like Automation Script (when a resource is deployed), custom deployment (deploy resources using common templates, Load templates from GitHub or deploy one from scratch).  
  
  
**Automation Script:**

From the Virtual Machine left blade, click Automation script. This is the easy way because we can view and download the template just by clicking some buttons.

**ADVANTAGES & DISADVANTAGES**

**Benefits of a 3-tier app architecture:**

The benefits of using a 3-layer architecture include improved horizontal scalability, performance and availability. With three tiers, each part can be developed concurrently by different team of programmers coding in different languages from the other tier developers. Because the programming for a tier can be changed or relocated without affecting the other tiers, the 3-tier model makes it easier for an enterprise or software packager to continually evolve an application as new needs and opportunities arise. Existing applications or critical parts can be permanently or temporarily retained and encapsulated within the new tier of which it becomes a component.

3-tier application programs may also be referred to as n-tier programs. In this context, the letter “n” stands for ‘a number of tiers.’

* **Scalability –**

The application server can be deployed on many machines.

Database no longer require connection from every client.

* **Data Integrity –**

Middle tier can ensure that only valid data is allowed to be updated in the database.

* **Improved Security –**

Because client does not have direct access to the database therefore data is more secure.

**Drawbacks of a 3-tier App Architecture:**

* The Three-tier nature makes it difficult for developers to change an application with the agility and flexibility they need to keep pace with the expectations of mobile users, and for operations teams to scale the application up and down to match demand.
* A Three-tier design hampers agility at several phases of the application development process. Even if application functionality is distributed in a modular fashion, a change to any module requires rebuilding and testing the entire application. This can be quite time-consuming. Nowadays, you need to roll out incremental improvements quickly and often to keep up with users’ thirst for ever-better performance and the latest cool new feature.
* The flexibility to choose from an array of solutions isn’t available with a 3-tier design, where solutions are typically built from a set of highly interdependent coupled components.
* The Three-tier architecture lacks scalability, it was designed in an era where the idea of elasticity and rapid scaling did not broadly exist. The functional components of the application are packaged together as a unit, because of which the only way you can respond to changing levels of client demand is to scale the entire application. Applications running on the three-tier architecture are typically unable to scale specific pieces of the application independently because the entire application is coupled together.

**HOW TO DEPLOY AN ARM TEMPLATE**

* In the following example suppose that we have already created an ARM template and are trying to deploy it.
* When we download the template.zip file and unzip it we can see the following files: **deploy.ps1**: A PowerShell script for template deployment.
* **deploy.sh**: A bash script that will deploy the ARM template.
* **deployer.rb**: A ruby script that will deploy the ARM template.
* **DeploymentHelper.cs**: A C# file with the template deployment.
* **parameters. json**: A JSON file that includes all the parameters for the deployment.
* **template. json**:  A JSON file that include the main deployment.
* After we do all the necessary configuration in the parameters. json and template.json files, we execute the following script.
* Login-AzureRmAccount.
* Set-AzureRmContext -SubscriptionId 'aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeee'
* New-AzureRmResourceGroup -Name ResourceGroupName -Location "West Europe"
* New-AzureRmResourceGroupDeployment -Name Example Deployment -ResourceGroupName ResourceGroupName `
* -Template File C:\MyTemplates\template\template.json `
* -TemplateParameterFile C:\MyTemplates\template\parameters.json

**3 - TIER ARCHITECTURE AND IBM CLOUD**

* Whether you need assistance with strategy, processes or capabilities—or want full-service attention—explore how IBM can help with [application modernization](https://www.ibm.com/in-en/cloud/application-modernization" \t "_blank" \o "application-modernization).
* Start using containerized middleware that can run in any cloud—all bundled in [IBM Cloud Pak’s](https://www.ibm.com/in-en/cloud/paks" \t "_blank" \o "cloud_paks).
* Easily migrate existing VMWare workloads to the public cloud using [IBM Cloud for VMware Solutions](https://www.ibm.com/in-en/cloud/vmware" \t "_blank" \o "vmware).
* Plan and execute your application modernization strategies as part of your ongoing digital transformation with help from IBM’s [application modernization services](https://www.ibm.com/in-en/services/cloud/modernize-applications" \t "_blank" \o "modernize-applications).

**INTERNSHIP CERTIFICATE**

****