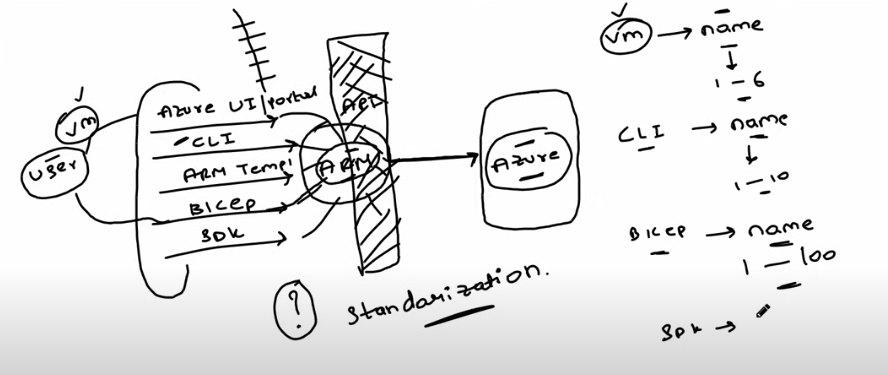
An **Azure Resource Manager (ARM) template** is a JSON file that defines the infrastructure and configuration for your Azure resources in a declarative way. Using ARM templates, you can automate the deployment and management of Azure resources, ensuring consistency, repeatability, and scalability in your infrastructure setup.



**Structure of an ARM Template:**

An ARM template typically consists of the following sections:

1. **Schema**: Defines the version of the ARM template format and validation.
2. **ContentVersion**: Specifies the version of your template.
3. **Parameters**: Enables input of customizable values during deployment.
4. **Variables**: Stores values that can be reused in the template.
5. **Resources**: Specifies the resources to deploy or update.
6. **Outputs**: Returns values after deployment (e.g., resource properties).

{

    "$schema": "https://schema.management.azure.com/schemas/2019-04-01/deploymentTemplate.json#",

    "contentVersion": "1.0.0.0",

    "parameters": {},

    "functions": [],

    "variables": {},

    "resources": [],

    "outputs": {}

}

{

    "$schema": "https://schema.management.azure.com/schemas/2019-04-01/deploymentTemplate.json#",

    "contentVersion": "1.0.0.0",

    "parameters": {

       "Strogename": {

        "type": "string",

        "defaultValue": "himanshu001",

        "metadata": {

            "description": "description"

        }

       }

    },

    "functions": [],

    "variables": {},

    "resources": [

        {

            "name": "[parameters('Strogename')]",

            "type": "Microsoft.Storage/storageAccounts",

            "apiVersion": "2023-04-01",

            "location": "[resourceGroup().location]",

            "kind": "StorageV2",

            "sku": {

                "name": "Premium\_LRS",

                "tier": "Premium"

            }

        }

    ],

    "outputs": {}

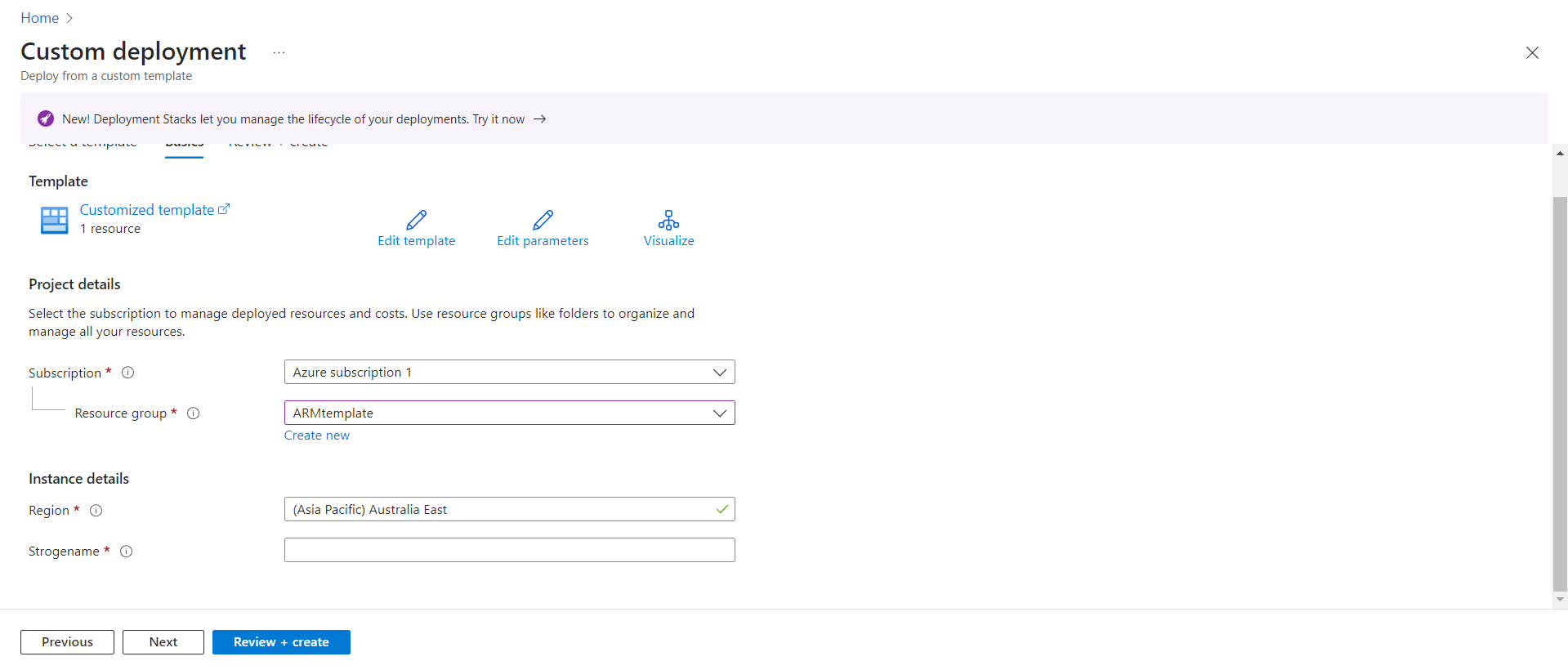
}

az group create --name ARMtemplate --location 'australiaeast'

az deployment group create -g ARMtemplate --template-file test.json

"name": "[uniqueString(resourceGroup().id)]",

**The use of "[uniqueString(resourceGroup().id)]" is valid for generating a deterministic unique name based on the resource group's ID**

****

**Cascading** in the context of Azure refers to designing or structuring your resources in a way that one resource depends on or is linked to another, creating a chain-like dependency. This is commonly used in **ARM templates** to ensure that resources are deployed in a specific sequence, as some resources depend on others to function properly.

**Example of Cascading in ARM Templates**

1. **Virtual Network (VNet)** is created first.
2. **Subnet** is created within the VNet.
3. **Network Security Group (NSG)** is associated with the subnet.
4. **Public IP and Network Interface (NIC)** are linked to the VNet.
5. Finally, the **Virtual Machine (VM)** is created and linked to the NIC.

In this scenario, each resource is **cascaded** to depend on the preceding one.

***Script to run in linux***

#!/bin/bash

# Variables

RESOURCE\_GROUP="ARM"

LOCATION="australiaeast"

TEMPLATE\_FILE="/home/sysadmin/Template/threetierarc.json"

# Create the resource group if it doesn't exist

echo "Checking if resource group exists..."

az group show --name $RESOURCE\_GROUP &>/dev/null

if [ $? -ne 0 ]; then

echo "Resource group $RESOURCE\_GROUP not found. Creating it..."

az group create --name $RESOURCE\_GROUP --location $LOCATION

else

echo "Resource group $RESOURCE\_GROUP already exists."

fi

# Validate the ARM Template

echo "Validating ARM Template..."

az deployment group validate --resource-group $RESOURCE\_GROUP --template-file $TEMPLATE\_FILE

# Deploy the ARM Template

echo "Deploying ARM Template..."

az deployment group create --resource-group $RESOURCE\_GROUP --template-file $TEMPLATE\_FILE

echo "Deployment completed!"