

WELCOME TO THE 30 DAYS OF AZURE TASKS

AUTHOR: Yatharth Chauhan (Github: YatharthChauhan2362)
TOPIC: 30 Days Of Azure Tasks

"30 Days of Azure Tasks" is likely a program or challenge that involves completing various tasks related to Azure over the course of 30 days. These tasks could include learning about different Azure services, building and deploying applications on Azure, or configuring and managing Azure resources.

Task-1

1. [Create a Vnet and set up a subnet on the created Vnet](#)
2. [Create a VM on the subnet and access VM](#)

Task-2

1. [Create Private and Public Subnets in the above created Vnet](#)
2. [Create a VM and attach multiple NIC with it](#)

Task-3

1. [Create multiple VMs \(i.e.\) VM1 and VM2 in two different Virtual networks.](#)
2. [Create a Network Security Group to establish connection among VMs to ping each other and understand how the priority works.](#)

Task-4

1. [Create 4 VNets](#)
2. [Management Vnet \(HUB\)](#)
3. [Production Vnet](#)
4. [Testing Vnet](#)
5. [Developing Vnet and Configure Hub and Spoke Architecture and verify it's working by launching VM in each VNet and ping from Management VM to every other VM.](#)

Task-5

1. [Learn about Load balancer and their types](#)
2. [Create Internal and External Load Balancer.](#)

Task-6

1. [Learn about VPNs and their types](#)
2. [Create Point to Site on Azure, A Point-to-Site \(P2S\) VPN gateway lets you create a secure connection to your virtual network from an individual computer](#)

Task-7

1. Create a Site-to-Site on Azure and explore the use case of Site-to-Site VPN gateway and figure out how Site-to-Site VPN is different from Point-to-Site VPN?

Task-8

1. Create a Vnet with 2 Subnets
2. Web-Tier: Linux VM, Windows VM.
3. Data-Tier: SQL DB
4. Launch VMS into Availability Set
5. Create an External and Internal Load Balancer and verify it is working.

Task-9

1. Create a VHD of an existing VM and store
2. it into Storage Account and using that VHD Create a new VM.

Task-10

1. Create a Vnet with single subnet and Create a VM.
2. Schedule a Daily backup of VM at 3:AM using vault
3. Create an Alert rule for VM CPU percentage:
4. Criteria: CPU% More Than 80%, There Should be an alert on email.

Task-11

1. Create a sample .NET application and host it on Azure App Services.
2. Create an application gateway and access Azure App service through Azure Application gateway.

Task-12

1. Launch a Linux VM and Install Docker services on that VM.
2. Launch a Docker container and set up an Apache/ Nginx server over that container.
3. Create a simple HTML page to launch over the web server.
4. Configure the Docker container such that when we start the VM, the Docker Services are up and running and the container is launched automatically such that the web server is turned to active status to reflect the web page just after Starting VM.
5. Perform the same task using Docker File

Task-13

1. Create a Bridge Network.
2. Launch two Docker Containers
3. Assign both containers with same network
4. Find out if this bridge network works like a switch between these two containers and like a router for any external IP.

Task-14

1. Mount a volume/ directory from the host system to a Docker container.

Task-15

1. Understand Kubernetes Architecture.
2. Launch a Linux VM
3. Setup Kubeadm on the Linux VM
4. Launch a Nginx pod and expose it using the NodePort and access the web server.
5. Difference between Kubeadm, Kubectl and Kubelet.

Task-16

1. Setup an Azure Kubernetes Services: Manual Scaling, Azure Vnet, Azure network policy
2. Deploy pods, and expose the pod using LoadBalancer service.
3. Perform the same task using Declarative approach as well.

Task-17

1. Create a replica set
2. Expose it using LoadBalancer service
3. Change the number replicas and verify
4. Remove the pods and verify if new pods are getting created automatically.
5. Perform the same task using Declarative approach as well.

Task-18

1. Create a Deployment and expose it using LoadBalancer service.
2. Set a different image for a new rollout.
3. Rollback to the previous deployment.

Task-19

1. Set up an organization over Azure DevOps services and create a public project (Note: Use Agile Process)
2. Create multiple work items and create hierarchy among these work items.
3. Create a sprint over Azure Boards.
4. R&D on distinguished scope of access to Azure DevOps Organization.

Task-20

1. Create a project with different user groups and implement group policies.
2. Apply branch policies that only project administrator can access master branch, and contributors cannot have access.

Task-21

1. Create a repository over Azure Repos with two separate branches.
2. Commit some basic application code on these branches with different Runtime Stack.
3. Deploy this application code onto two different Azure App services by setting up Azure DevOps CI/CD pipelines for both deployments (Note: Using Classic editor).

Task-22

1. Deploy a .NET microservice application on Kubernetes service using Azure pipelines.

Task-23

1. Apply security on branches that contributors can create a pull request but cannot directly merge code to master.

Task-24

1. Apply approval gates in pipelines
2. Apply branch filters and path filters

Task-25

1. Use work items in pipelines.

Task-26

1. Use Pipeline variable while configuring pipelines.

Task-27

1. Use variable and task groups in pipelines and set scopes for different stages

Task-28

1. Create a containerized web app for .NET/Python/Java/Node.js (Note: Use GitHub as a SCM tool)
2. Deploy the application code on Azure Kubernetes Services.

Task-29

1. Coming Soon

Task-30

1. Coming Soon