Jacob Brown

Phoenix cloud Solutions | jacksonville, FL

Digital Doc

Phoenix Cloud solutions

Jacksonville, FL

Table of Contents

[Project Title 2](#_Toc65318579)

[Project Overview 2](#_Toc65318580)

[Project Goals & Objectives 2](#_Toc65318581)

[Network Infrastructure Requirement 3](#_Toc65318582)

[Assumptions & Risks 3](#_Toc65318583)

[Network Configuration & Procedural Checklist 3](#_Toc65318584)

[Network Topology Diagram with IP Addresses 4](#_Toc65318585)

[Appendices | Deliverables 5](#_Toc65318586)

[Azure 5](#_Toc65318587)

[Create a Resource Group 5](#_Toc65318588)

[Create a Virtual Network 6](#_Toc65318589)

[Create Virtual Machines 8](#_Toc65318590)

[Ansible Installation and Setup 13](#_Toc65318591)

[Ansible is installed on macOS using the Python package manager ‘pip.’ 13](#_Toc65318592)

[Obtain project scripts from GitHub. 13](#_Toc65318593)

[File explanations. 14](#_Toc65318594)

[Ansible Script Prep 15](#_Toc65318595)

[Gather Info 15](#_Toc65318596)

[Updating Ansible Scripts 16](#_Toc65318597)

[Initial SSH Connection 18](#_Toc65318598)

[Run Ansible Scripts 19](#_Toc65318599)

[Create the database 20](#_Toc65318600)

[Testing Setup 21](#_Toc65318601)

[Secure Connection 21](#_Toc65318602)

[Persistence, Load Balancing, Failover 22](#_Toc65318603)

[HAProxy Stats Page 23](#_Toc65318604)

**Scope of Work**

# Project Title

Digital Doc: Document Management System with Three-Tier Architecture

# Project Overview

This project is to create a document management system that is cloud-based. The three tiers will include the database, web application, and load balance layers. The load balancer that is accessible to the user is HAproxy. HAproxy will connect the users to one of three SeedDMS web application servers. The SeedDMS web servers will connect to the Maria database, where the files will be kept. Secure encrypted connections will be used and tested for redundancy and failover. The project completion date is February 28, 2021.

Milestone completion dates:

Milestone I – February 7, 2021

Milestone II – February 14, 2021

Milestone III – February 21, 2021

Milestone IV – February 28, 2021

# Project Goals & Objectives

With the completion of this project, the company should see the following improvements.

* **Security** - The three-tier architecture and three-tier network will funnel connections to the correct ports and block all other requests. The load balancer is the only public-facing asset, and the web application servers make the requests to the database layer.
* **Scalability** - The tiered architecture allows scaling out and back in to meet peak and off-peak loads without making changes to the design. The tiered network currently can accommodate ten load balancers, 120 web application servers, and ten database servers in addition to this project.
* **Flexibility** - The architecture also allows for testing updates or new software versions by enabling the ability to switch back to the last known configuration. Scaling up and down is possible by bringing online larger servers during peak times and scaling down to smaller servers during off-peak hours.
* **Reliability** - Azure has a 99.99% uptime for two or more virtual machines, and locally redundant storage is 99.999…% (11 nines) of durability. The tiered architecture will allow for failures in the web applications layer and still provide functionality.

# Network Infrastructure Requirement

Users need to have access to a computer with an active internet connection to utilize the Seed Document Management System (SeedDMS).

# Assumptions & Risks

* The company has a basic understanding of Azure and a Service Level Agreement (SLA).
* The company will have administrators assigned to manage users’ permissions by granting and revoke access as people leave and join the company or change positions.
* The company will provide training to employees on SeedDMS functionalities, company record-keeping for regulatory requirements, and critical corporate documents.
* The company must provide access from the company’s infrastructure with internet access to SeedDMS. This is typically done with direct access on-premises or access to the on-premises network via a Virtual Private Network (VPN).
* The company must keep and review server monitoring and logging data for system health and provisioning. Server updates and software upgrades must be performed for the security and functionality of the architecture.

# Network Configuration & Procedural Checklist

1. Setup Azure Environment
2. Provision VMs
3. Install Ansible
4. Prep Scripts
5. Run scripts for Database, Web Application, and Load Balancer
6. Test configurations

# Network Topology Diagram with IP Addresses

Diagram

Description automatically generated

# Appendices | Deliverables

## Azure

Here we will create an Azure Resource Group, virtual networks, and virtual machines. We could build everything we need during the creation of a Virtual machine, but that would give us a flat network when we want a segmented network. A segmented network will give added communication controls for the three-tier architecture. The goal is to prevent public access to anything but the load balancer and only accept internal requests from specific networks.

### Create a Resource Group

A picture containing text, screenshot, monitor

Description automatically generated

1. Enter ‘Resource Groups’ into the search field.
2. Click ‘Create’
3. Group fields:
   1. Select Subscription
   2. Enter the new Resource Group name
   3. Select Region
4. Click ‘Review + Create’ and ‘Create.’

### Create a Virtual Network

**Basics page**  
A picture containing text, screenshot, monitor

Description automatically generated

1. Enter ‘Virtual Network’ into the search field.
2. Click ‘Create’
3. Group fields:
   1. Select Subscription
   2. Select Resource Group we created earlier.
   3. Enter the name of the new network
4. Click ‘Next: IP Addresses’

**Virtual Network Address Spaces**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Functions** | **Number of Devises** | **Total Number of Hosts** | **Network ID** | **IP Address Range** | **Broadcast ID** | **CIDR Notation** | **Available Addresses (-5 Azure)** |
| IPv4 Address Space | N/A | N/A | 10.33.105.0 | 10.33.105.1 - 10.33.105.254 | 10.33.105.255 | /24 | N/A |
| Web App | 3 | 128 | 10.33.105.0 | 10.33.105.1 - 10.33.105.126 | 10.33.105.127 | /25 | 120 |
| Database | 1 | 16 | 10.33.105.128 | 10.33.105.129 - 10.33.105.142 | 10.33.105.143 | /28 | 10 |
| Load Balancer | 1 | 16 | 10.33.105.144 | 10.33.105.145 - 10.33.105.158 | 10.33.105.159 | /28 | 10 |
| OPEN | 0 | 16 | 10.33.105.160 | 10.33.105.161 - 10.33.105.174 | 10.33.105.175 | /28 | 11 |
| OPEN | 0 | 16 | 10.33.105.176 | 10.33.105.177 - 10.33.105.190 | 10.33.105.191 | /28 | 11 |
| OPEN | 0 | 64 | 10.33.105.192 | 10.33.105.193 - 10.33.105.254 | 10.33.105.255 | /26 | 59 |

**IP Addresses Page**  
Graphical user interface, application

Description automatically generated

1. Enter the IP Address Space: 10.33.105.0/24
2. Select ‘Add Subnet’:
   1. Enter Subnet Name: LoadBalancer
   2. Enter Subnet address: 10.33.105.144/28
   3. Select ‘Save’

NOTE: Repeat these steps for WebApp:10.33.105.0/25

NOTE: Repeat these steps for Database:10.33.105.128/28

1. After creating all three subnets, select ‘Review + Create’ and ‘Create.’

### Create Virtual Machines

**Basics Page**  
Graphical user interface, application

Description automatically generated

1. Enter ‘Virtual Machines’ into the search field.
2. Click ‘Create’
3. Group fields:
   1. Select the Resource Group we created earlier.
   2. Enter Database for the VM name.

Enter SeedDMS-1 through 3 for WebApps.

Enter LoadBalancer for the load balancer.

* 1. Select the ‘CentOS-based 8.2 - Gen1’ image.

For the LoadBalancer, Select the ‘Ubuntu Server 18.04 LTS - Gen1’.

* 1. Select the ‘Standare\_B1s - 1 vcpu, 1 GiB memory’ size

NOTE: Scroll the Azure blade to see the second part.

**Basics Page (continued)  
 Graphical user interface, text, application

Description automatically generated**

We are going to use the same SSH public key generated here for all the VMs

1. SSH Public Key
   1. Enter ‘azureuser.’
   2. Select ‘Generate new key pair.’
   3. Enter ‘seeddms\_key.pem.’
2. Click ‘Next: Disks’

**Disks Page**Graphical user interface, text, application

Description automatically generated

1. Select ‘Standard HDD’
2. Select ‘Next: Networking’

**Networking Page  
Graphical user interface, application

Description automatically generated**

1. Group Field:
   1. Select the ‘Database’ Subnet

NOTE: This will be WebApp or LoadBalancer for other VMs.

* 1. Select ‘(new)’ for Public IP
  2. For the Database, VM Select port ’22 and 3389” for inbound ports.

NOTE: For WebApp and LoadBalancer VMs, Select ports ’22, 80, and 443” for inbound ports.

1. Select ‘Review + Create’ and ‘Create.’

**Download the new key pair.**Graphical user interface, application

Description automatically generated

1. Select “Download private key and create resources.”

A screenshot of a computer

Description automatically generated with medium confidence

1. Save the new key pair.
   1. Location/name: ~/Azure\_Pem/Seeddms\_key.pem

Repeat the ‘[Create Virtual Machine](#_Create_Virtual_Machines)’ for the other four VMs. The WebApp servers are the same as above, except the Subnet group is WebApp. The LoadBalancer uses Ubuntu Server 18.04 for the OS, and the Subnet group will be the LoadBalancer subnet. All the VMs use the same SSH key for access.

## Ansible Installation and Setup

This tutorial will explain how to install Ansible on a macOS version 10.12 or newer and assumes Ansible has never been install before. If you are installing on a different operating system or need more install help due to more recent versions, please see: [Ansible Installation Guide](https://docs.ansible.com/ansible/latest/installation_guide/intro_installation.html)

### Ansible is installed on macOS using the Python package manager ‘pip.’

1. Install pip:
   1. curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py
   2. python get-pip.py --, user
2. Install Ansible.
   1. python -m pip install --user Ansible
3. Install ‘paramiko.’
   1. python -m pip install --user paramiko

### Obtain project scripts from GitHub.

1. Make a working directory:
   1. mkdir ~/Project\_Portfolio\_5
   2. cd ~/ Project\_Portfolio\_5
2. Download scrips from GitHub.
   1. git clone <https://github.com/Brownj0125/Project_Portfolio_5.git>

### File explanations.

The ‘ansible.cfg’ file is a one-liner that tells Ansible to look for a local host file (line 14). The ‘hosts’ file is the who, where, and how Ansible will connect to your Azure VMs. The ‘MariaDB.yml,’ ‘SeeddmsPB.yml,’ and ‘HAProxy.yml’ files are the Ansible Playbooks that direct the updates, installations, and configuration changes of the VMs. The items in the ‘Files’ directory are the configured files that replace the system defaults. The Playbooks will be explained later in the instructions.

**Host File**Text

Description automatically generated

[MariaDB] - This is the name of a list of IP addresses for Playbooks to call on.

[MariaDB:vars] - Are the variables that apply to all the hosts. I defined the username and the location of the .pem file we will create later in Azure.

## Ansible Script Prep

First, you will need to update the script to change IP address, file locations, usernames, and passwords. Next, we will establish an SSH connection with each of the servers.

### Gather Info

A picture containing text, monitor, indoor, open

Description automatically generated

1. Enter ‘Virtual Machines’ into the search field.
2. Click the first VM name
3. Select ‘Overview’
4. Write down the Public and Private IP addresses for each server.

### Updating Ansible Scripts

1. Update the Ansible host file with all public IP and point Ansible to the Azure “.pem” file.
   1. Location: ~/Project\_Portfolio\_5/hosts

Graphical user interface, text

Description automatically generated with medium confidence

1. Update the HAproxy file with the WebApp’s private IP Address.
   1. Location: ~/Project\_Portfolio\_5/Files/haproxy.cfg (Lines: 47-49)

Text

Description automatically generated

1. Update the SeedDMS ‘setting.xml’ file with the private IP address. You can change the SQL database name, username, and password, but it will need to match the MariaDB.yml playbook.
   1. Location: ~/Project\_Portfolio\_5/Files/settings.xml (Line: 101)



* 1. Location: ~/Project\_Portfolio\_5/MariaDB.yml (Lines: 26 - 38)

Text

Description automatically generated

1. Update the file location to match your local directory setup.
   1. Location: ~/ Project\_Portfolio\_5/SeeddmsPB.yml (Line: 66)

Text

Description automatically generated

* 1. Location: ~/ Project\_Portfolio\_5/HAProxy.yml. (Line: 97)

Text

Description automatically generated

* 1. Location: ~/ Project\_Portfolio\_5/HAProxy.yml. (Lines: 58 - 71)

Text

Description automatically generated

### Initial SSH Connection

Before you can run the Ansible Playbook, you need to connect to the servers. You will need to open Terminal on your macOS

Text

Description automatically generated

1. Move to the Ansible project file:
   1. cd ~/Project\_Portfolio\_5
2. Use SSH and the .pem file to connect to the VM’s public IP address:
   1. ssh -I ~/Azure\_Pem/seeddms\_key.pem azureuser@<PublicIP>
3. When prompted, enter ‘yes.’
4. Enter ‘exit’ to go back to your terminal session.

Repeat these four steps for each server.

## Run Ansible Scripts

1. Run command: ansible-playbook MariaDB.yml
   1. You have an updated and configured database server.
2. Run command: ansible-playbook Seeddms.yml
   1. You have three updated and configured SeedDMS web application servers
3. Run command: ansible-playbook HAProxy.yml
   1. You have an updated and configured load balancer with SSL/TLS termination

|  |  |  |
| --- | --- | --- |
| MariaDB | SeedDMS | HAproxy |
| Text  Description automatically generated | A picture containing text, electronics, computer  Description automatically generated | Text  Description automatically generated |

### Create the database

1. Open your browser to your Load Balancer’s IP and add /install/install.php



1. Check the ‘Create database tables’ and click ‘Apply.’

Graphical user interface, text, application

Description automatically generated

1. Back in Terminal, run Ansible playbook to clean up the database initiation.
   1. Run command: ansible-playbook VMCleanup.yml

A picture containing text, screenshot, black

Description automatically generated

## Testing Setup

### Secure Connection

To check the encryption of the load balancer. Use your browser and enter the web address in the address bar.

1. After the page is displayed, you right-click the lock icon (pictured is Not Secure)
2. select certificate.
   1. The Public Key Info is the encryption used for the connection

|  |  |
| --- | --- |
| Graphical user interface, website  Description automatically generated | Graphical user interface, application  Description automatically generated |

### Persistence, Load Balancing, Failover

Currently, there is a cookie that keeps your session locked into one server. Once you logout and close your browser, the cookie is removed. To test this, I will shut down the server.

1. We are locked into SeedDMS-3

Graphical user interface, text, application, email

Description automatically generated

1. Now I’ll shut down the SeedDMS-3 server from Azure.

Graphical user interface, application

Description automatically generated

1. Back to the SeedDMS control panel and refresh the browser page.
   1. The load balancer failed over to SeedDMS-2.
   2. The cookie kept the session alive, and I did not need to log in again.

Graphical user interface, text, application, website

Description automatically generated

### HAProxy Stats Page

This page shows the numbers the HAProxy captures and health assessments of the SeedDMS and backend servers.

Graphical user interface, text, application

Description automatically generated