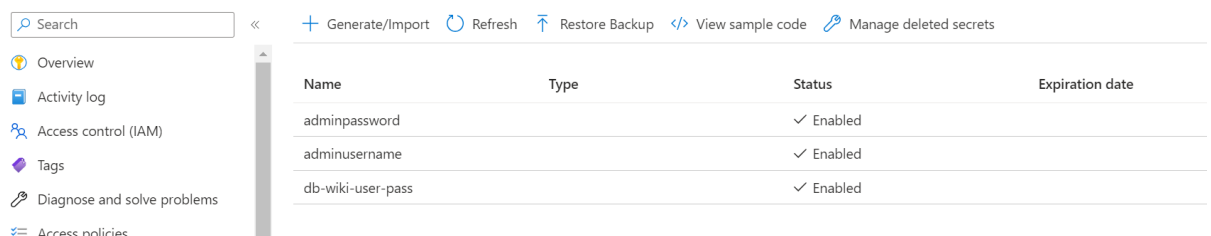


## Mediawiki deployment using Terraform and Ansible integrated with azure DevOps Pipeline

This document contains the steps required to deploy mediawiki over Apache server in Redhat virtual machine in azure.

Steps:-

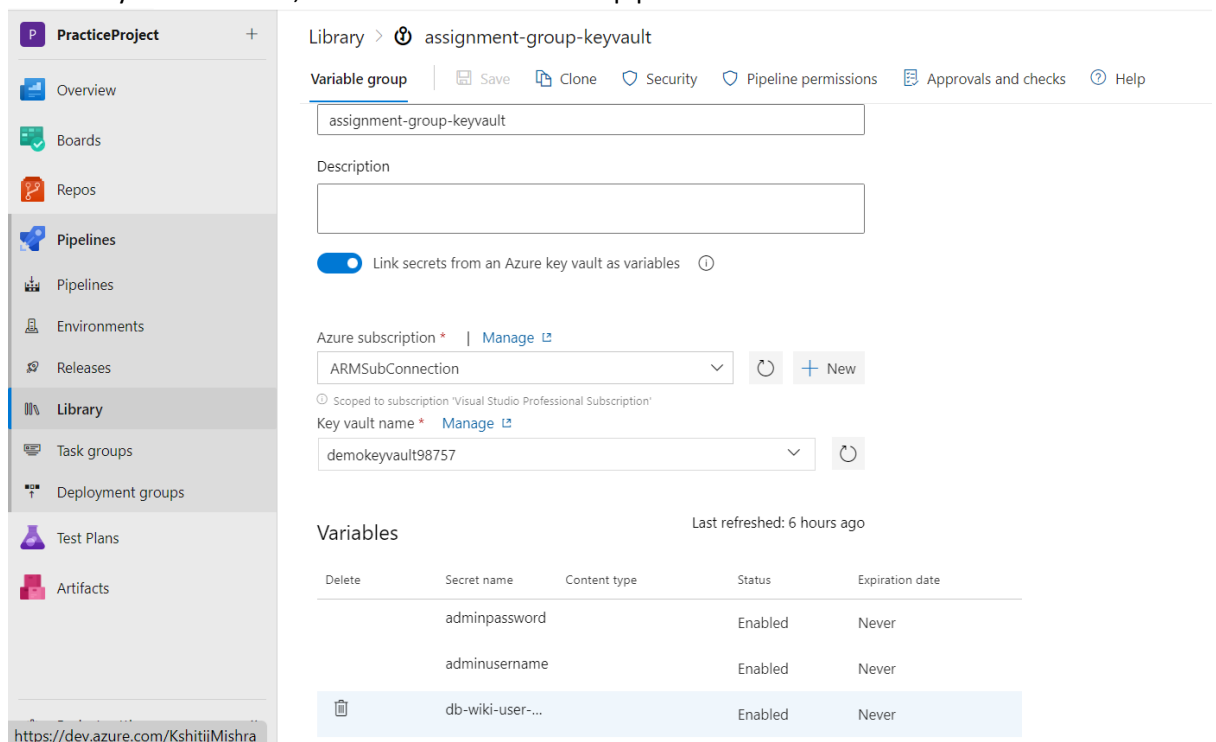
1. Put the code files in azure repos as that is the default code repo for azure DevOps and all pipelines has been created using azure repos integration.  
Note: - You can use github repo as well but that requires additional layer of authentication.  
So, this pipeline code mainly supports azure repos as of now.
2. Create a service connection with azure subscription with contributor role. Also, create one keyvault and provide the get and list permission to service connection's service principle over keyvault . Create the below secrets in keyvault which can be used as sensitive variable by terraform and ansible in pipelines.



The screenshot shows the Azure Key Vault interface. On the left is a navigation pane with options: Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, and Access policies. The main area has a search bar and action buttons: Generate/Import, Refresh, Restore Backup, View sample code, and Manage deleted secrets. Below these is a table of secrets.

Name	Type	Status	Expiration date
adminpassword		✓ Enabled	
adminusername		✓ Enabled	
db-wiki-user-pass		✓ Enabled	

3. Now, create variable group named assignment-group-keyvault and connect it to previously create keyvault . So that, we can use secrets in our pipeline.



The screenshot shows the 'assignment-group-keyvault' variable group configuration in Azure DevOps. The left sidebar shows the 'Library' section. The main area has tabs for 'Variable group', 'Save', 'Clone', 'Security', 'Pipeline permissions', 'Approvals and checks', and 'Help'. The 'Variable group' tab is active, showing the name 'assignment-group-keyvault' and a description field. Below this is a toggle for 'Link secrets from an Azure key vault as variables' which is turned on. The 'Azure subscription' is set to 'ARMSubConnection'. The 'Key vault name' is 'demokeyvault98757'. At the bottom, a 'Variables' table lists secrets from the key vault.

Library > assignment-group-keyvault

Variable group | Save | Clone | Security | Pipeline permissions | Approvals and checks | Help

assignment-group-keyvault

Description

☐ Link secrets from an Azure key vault as variables ⓘ

Azure subscription \* | Manage [Manage](#)

ARMSubConnection

ⓘ Scoped to subscription 'Visual Studio Professional Subscription'

Key vault name \* [Manage](#)


demokeyvault98757







Variables Last refreshed: 6 hours ago

Delete	Secret name	Content type	Status	Expiration date
	adminpassword		Enabled	Never
	adminusername		Enabled	Never
<input type="button" value="Delete"/>	db-wiki-user-...		Enabled	Never

<https://dev.azure.com/KshitijMishra>

4. Create a storage account and a blob container (with Azure Active directory authentication method). Assign blob data contributor role to service principal which is created in above steps. This storage account container will be used to store terraform state file.
5. Now, Create a variable group named assignment-group in azure devops and storage account details, subscription details and terraform version and etc. Please refer the variable from below snip, keep the variable names same and put the variable values as per your environment.

Library >  assignment-group

Variable group |  Save |  Clone |  Security |  Pipeline permissions |  Approvals and checks |  Help

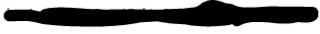
### Properties

Variable group name

Description

☒ Link secrets from an Azure key vault as variables ⓘ

### Variables

Name ↑	Value	🔒
backend-container-name	statecontainer	
backend-rg-name	practiceRg	
backend-storage-name	assignmentstorage56748	
backendsubscriptionid		
public_ip_name	assignment_vm_public_ip	
rg_name	assessme_rg	
terraform_version	latest	

6. Finally, Create a pipeline with azure-pipelines.yaml file (stored in repo in root directory) and run that pipeline. After completion of pipeline, Vm will be created and mediawiki will be deployed. Check the VM's public ip in the subscription and hit it in the browser.

Note: - This is a simple architecture. There are lots of option available for security and networking like environments in azure DevOps, approval and checks, linting, application gateway, ansible vault, VMSS (For scaling and high availability purpose), Azure Firewall, DNS zones for providing dns name to IPs etc. It was not possible to implement all in such short time.