## Provision Azure File Share with Private Endpoint and Storage Account

- Overview
- Problem
- Prerequisites
- Solution
- · Related articles



Platform:	Azure
Owner of this SOP:	Fully Managed POD A
Cloud Services:	File Share Provisioning



When requesters are asking for creating Azure File share with Private Endpoint and Storage account.

tester	Reviewer
@ Ramkumar Samudram (Deactivated)	

## Prerequisites

- Install VS Code
- Access to Terraform workspace and azure portal

## **Solution**

- Clone the Blue Print 10 repo "https://gitlab.com/bhp-cloudfactory/azure-blueprints/azr-file-share" into your repository under new project e.g. htt ps://gitlab.com/bhp-cloudfactory/tooling-foundations/fm-sops/azrfileshare

  Change all the variables in the tfvars file to suit your application

```
回の哲却
∨ AZRFILESHARE
                                       env_vars > "npe.auto.tfvars
                                               # Storage Account
 > doc
                                                                          = "it"
                                              base name

✓ env_vars

                                              name
                                                                          = "fmpodastorage"
  rpe.auto.tfvars
                                                                          = "it-aue1-npe-arg-fmpodafileshare"
                                              resource_group_name
 scripts
                                                                                                   # set this to fals
                                              create_resource_group
                                                                          = true
  env.sh
                                              resource_group_lock_level = ""
                                                                                                    # Specifies the Le
  fetch secrets.sh
                                                                          = "Australia East"
                                               location
                                                                          = "Standard" # Account tier
  ■ file_share.sh
                                               account_tier
                                               account_replication_type = "LRS" # Replication type
  seed_creds.py
                                               storageid_log_dest
                                                                         = "/subscriptions/01818878-c294-4617-a15a
  ■ terraform-apply.sh
  ■ terraform-destroy.sh
                                               #key-vault information for data block
  ■ terraform-plan.sh
                                               key_vault_name = "it-fmpoda"
  update_workspace.py
                                              key_vault_rg = "it-aue1-npe-arg-operations"
                                        14
  update_workspace.sh
  ■ validate.sh
                                               # Endpoint
                                               storage_endpoint_subnet = {

✓ templates

                                                                      = "sub-iaas"

■ configurersion.json.template

                                                 resource_group_name = "it-aue1-npe-arg-network"

    ■ destroy.json.template

                                                 virtual network name = "it-aue1-npe-vnt-10.125.121.0"

≡ envvariable.json.template

    □ run.json.template

                                               # Tags
  ■ uploadstate.json.template
                                               tags = {

    □ variable_new.template

                                                  BusinessUnit = "Fully Managed POD A"
 ! .gitlab-ci.yml
                                                  CostCentre = "1410.C.50048.0001"
 ! .terraform-docs.yaml
                                                  Env
                                                                 = "NPE"
 🦖 data.tf
                                                                = "danny.seow1@bhp.com"
 locals.tf
                                                  SecondaryOwner = "ayush.tripathi@bhp.com"
                                                  DeploymentType = "automated"
 w meta.tf
                                                  SolutionName = "toolingdev"
> OUTLINE
                                                                 = "Testing"
                                                  RequestID
> TIMELINE
```

• Change the variables in the env.sh file to match your environment

```
√ AZRFILESHARE

  > doc
                                                        export SUBSCRIPTION_ID="01818878-c294-4617-a15a-8b7934a433c4"

✓ env vars

   mpe.auto.tfvars

✓ scripts

                                                       export TFB_ORG="BHPEG-Dev"
    env.sh
  fetch_secrets.sh
                                                        #export TFC ORG="BHPEG"
  file share.sh
   seed_creds.py
   terraform-apply.sh
                                                 16 export CLOUD_SERVICE_PROVIDER="m" # 'a' for Amazon Web Service, 'm' for Microsoft Azure
17 export ASSET="it" # it (IT), bma (BMA), bmc (BMC), nswec (NSWEC), niw (Nickel West), od (Olympic Dam)

    ■ terraform-destroy.sh

   terraform-plan.sh
   update workspace.py
   update_workspace.sh
                                                      # "North Central US" = "usnc" // Illinois
# "South Central US" = "ussc" // Texas
# "West US 2" = "usw2" // Washington state (seattle)
    validate.sh

✓ templates

    ≡ configurersion.json.template

■ destroy.json.template

                                                        export REGION="aue1"

    ≡ envvariable.json.template

                                                       export ENV=${ENVIRONMENT}

    □ run.json.template

                                                 28 export TFC_ADDR="app.terraform.io"
   ■ uploadstate.json.template
   ■ variable_new.template
  ! .gitlab-ci.yml
                                                      export TFC WORKSPACE NAME="a azr toolingdev npe fmpodastorage 01818878-c294-4617-a15a-8b7934a433c4"
     .terraform-docs.yaml
                                                       export TFC_VERSION="1.0.3"
  data.tf
```

• In fetch\_secrets.sh update "your\_tfc\_token\_name" to keyvault variable which has your token

```
AZRFILESHARE
                    C: C ひ 母 scripts > ■ fetch_secrets.sh
env.vars
                                            if [ "$client_id" == "" ]
npe.auto.tfvars
                                                az login --identity --allow-no-subscriptions --output none
env.sh
 fetch_secrets.sh
                                                az login --service-principal --username $client_id --password "$client_secret" --tenant $tenant_id --outp
file_share.sh
seed_creds.py
                                           export client_id=`az keyvault secret show --name sp-app-id --vault-name it-fmpoda | jq -r .value`
terraform-apply.sh
                                           export client_secret=`az keyvault secret show --name sp-app-secret --vault-name it-fmpoda | jq -r .value`
terraform-destroy.sh
 terraform-plan.sh
                                           export TFB_TOKEN='az keyvault secret show --name tfb-token --vault-name it-fmpoda | jq -r .value'

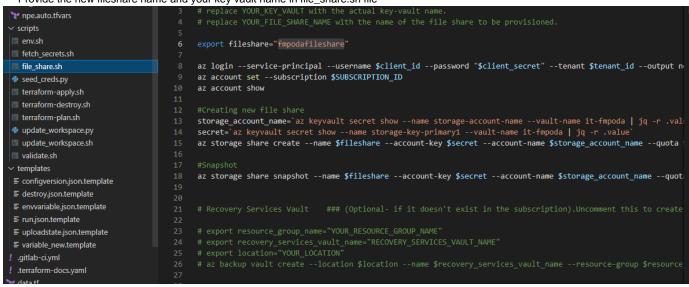
    update_workspace.py

update_workspace.sh
validate.sh
                                           export TFC_NSPACE_TOKEN='az keyvault secret show --name TFC-NSPACE-TOKEN --vault-name it-akv-001 | jq -r .val

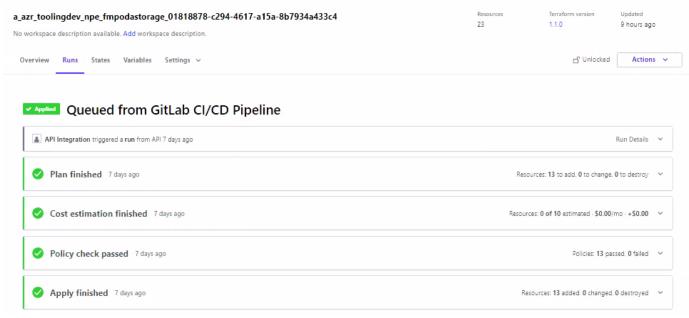
    templates

F configuersion.json.template
```

Provide the new fileshare name and your key vault name in file\_share.sh file



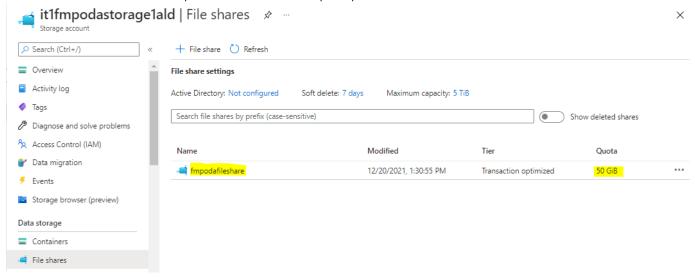
- Commit and push your changes
- Verify that the pipeline executes as expected and that a terraform workspace is created and run
- Verify the terraform plan in your TF workspace
- If you are satisfied with the plan output, initiate the apply stage in pipeline



• Manually start create\_share stage in pipeline after successful creation of storage account.



Check and confirm whether the required fileshare is created as per requirement



## Related articles

- Mount the Azure file share Cloud Platform Confluence (atlassian.net)
- FileShare soft delete Cloud Platform Confluence (atlassian.net)
- Schedule backups on Azure File Share Cloud Platform Confluence (atlassian.net)
- File Share Files restore from snapshot or backup Cloud Platform Confluence (atlassian.net)