

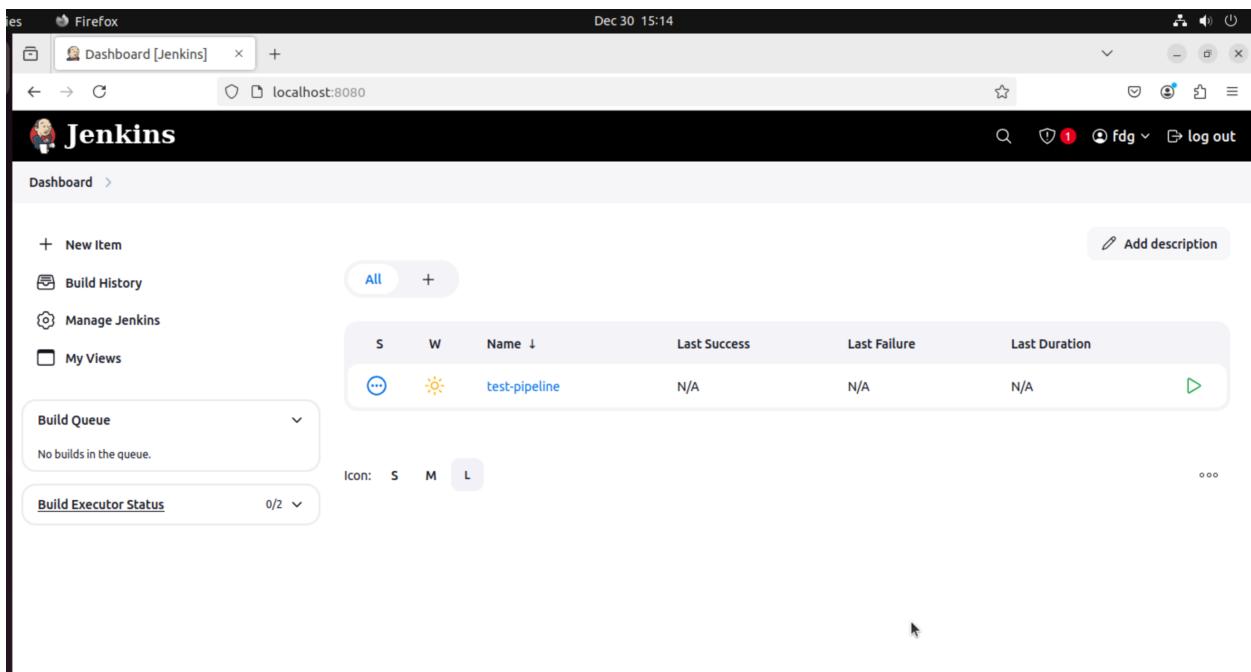
# Deploying a virtual machine to Azure using Terraform Jenkins CI/CD

In this section, we will create Jenkins pipelines to deploy our infrastructure to Azure Portal. In our Terraform scripts, we would like to create a Virtual machine and its resources on Azure. To achieve this goal first we would like to install Jenkins on our Ubuntu system.

```
docker pull jenkins/jenkins
docker create volume jenkins-volume
docker run -d -v jenkins-volume:/var/jenkins_home -p 8080:8080
```

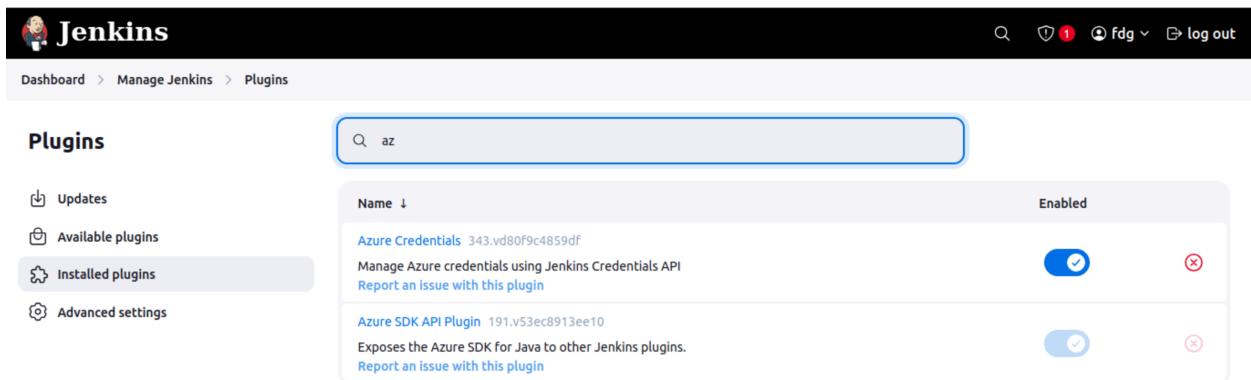
We can see that our jenkins now runs in a docker container;

```
root@dazzle-virtual-machine:/home/dazzle# docker ps
CONTAINER ID   IMAGE       COMMAND       CREATED      STATUS      NAMES
          PORTS
eaeca3b4e0ba   jenkins/jenkins   "/usr/bin/tini -- /u..."   5 hours ago   Up 5 hours   funny_jennings
rs  0.0.0.0:8080->8080/tcp, :::8080->8080/tcp, 50000/tcp
```

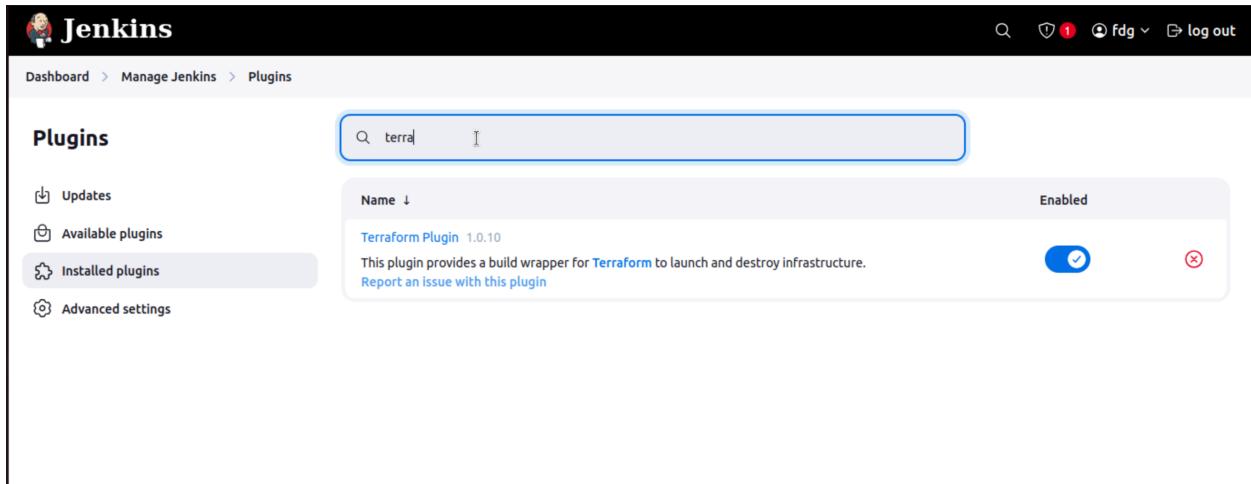


The screenshot shows the Jenkins Dashboard. On the left, there is a sidebar with links: '+ New Item', 'Build History', 'Manage Jenkins', and 'My Views'. Below these are two sections: 'Build Queue' (No builds in the queue) and 'Build Executor Status' (0/2). The main area displays a table for the 'test-pipeline'. The table has columns: S (Status), W (Workflow), Name (test-pipeline), Last Success (N/A), Last Failure (N/A), and Last Duration (N/A). There is a 'D' icon in the last column. At the bottom of the table, there are icons for S, M, and L, and a 'ooo' link.

We need to install our plugins to Jenkins such as Terraform and Azure Credentials;

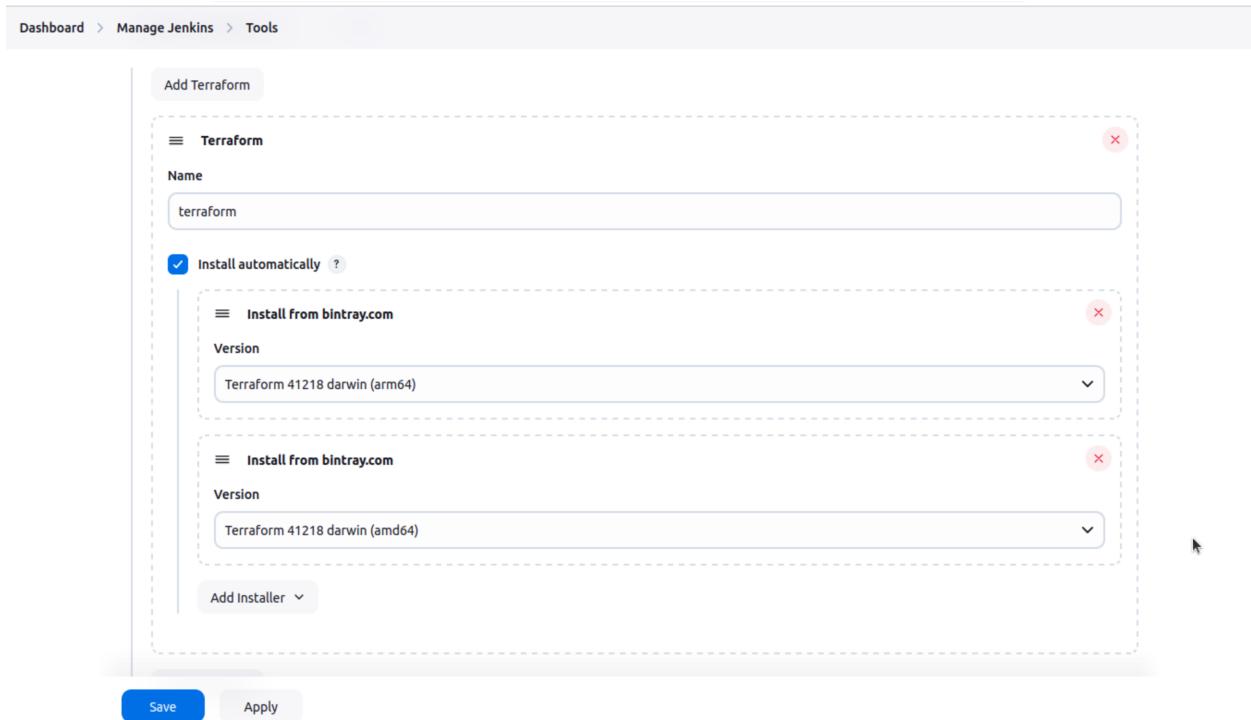


The screenshot shows the Jenkins Manage Jenkins > Plugins page. The left sidebar has links: 'Updates', 'Available plugins', 'Installed plugins' (which is selected and highlighted in grey), and 'Advanced settings'. A search bar at the top right contains the text 'az'. The main table lists installed plugins: 'Azure Credentials' (version 343.vd80f9c4859df, Enabled) and 'Azure SDK API Plugin' (version 191.v53ec8913ee10, Enabled). Each plugin has a 'Report an issue with this plugin' link.



The screenshot shows the Jenkins Plugins page. A search bar at the top contains the text 'terra'. Below it, a table lists the 'Terraform Plugin' version 1.0.10, which is described as providing a build wrapper for Terraform to launch and destroy infrastructure. The plugin is marked as 'Enabled' with a green checkmark and a red 'x' icon for unabling.

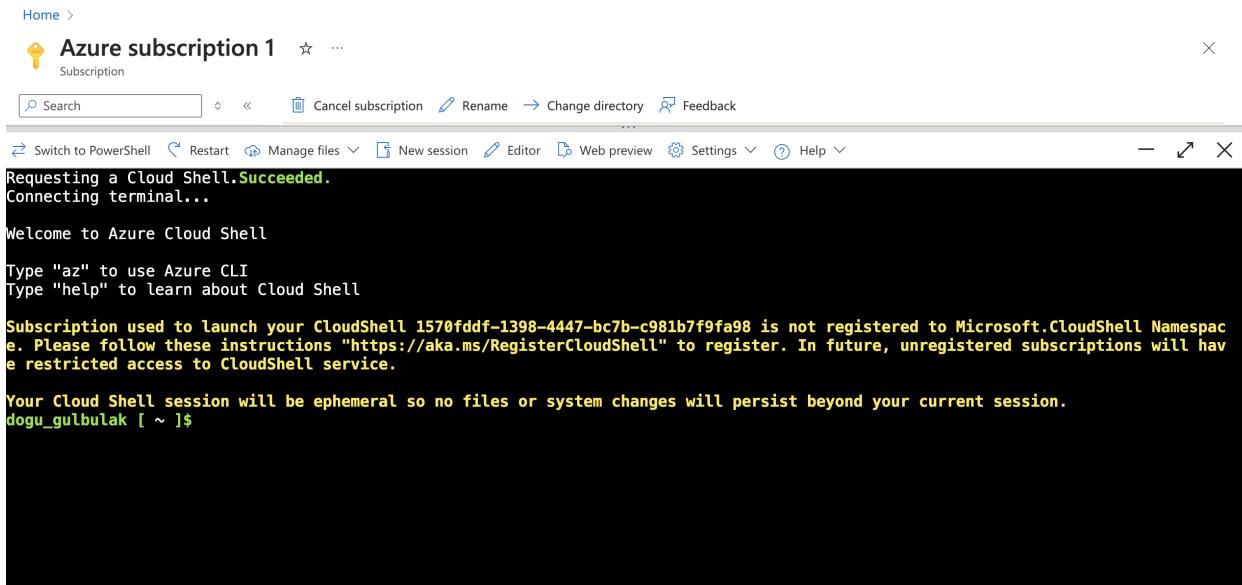
We need to add Terraform tool;



The screenshot shows the Jenkins Tools page. A configuration panel for 'Terraform' is displayed. The 'Name' field is set to 'terraform'. The 'Install automatically' checkbox is checked. Under the 'Install from bintray.com' section, the 'Version' dropdown is set to 'Terraform 41218 darwin (arm64)'. There is another identical section below it. At the bottom of the panel are 'Save' and 'Apply' buttons.

Now we would like to create an Azure Service Principle to use RBAC to provision our infrastructure (owner or contributor role).

Let's move over to Azure CLI;



```
az account show --subscription 1570fdddf-1398-4447-bc7b-c981b7f9fa98

az ad sp create-for-rbac --name "jenkins-test-sp" --role contributor

#If you want to assign new permissions to a service principal for a specific resource group

az role assignment create --assignee cd15bdcaa-5a08-4440-a8af-bab2e4d07 --scope /subscriptions/1570fdddf-1398-4447-bc7b-c981b7f9fa98/resourceGroups/test-jenkins
```

```
firat [ ~ ]$ az ad sp create-for-rbac --name "jenkins-test-sp" --role contributor --scopes /subscriptions/ba38195f-caaf-46fd-8149-0ee5f6d5b2cc/resourceGroups/test-jenkins
Found an existing application instance: (id) a5d9c9ac-8650-4a8e-b8d1-6f925f136469. We will patch it.
Creating 'contributor' role assignment under scope '/subscriptions/ba38195f-caaf-46fd-8149-0ee5f6d5b2cc/resourceGroups/test-jenkins'
{
  "appId": "cd15bdcaa-5a08-4440-a8af-bab2e4d07",
  "displayName": "jenkins-test-sp",
  "password": "en38Q~aNTSDPAD254qIBUsJoR8XL0Nc2K3Yngafq",
  "tenant": "f6eea791-8ab1-4047-9408-b47ff76cfda9"
}
```

We will use these to create a secret inside Jenkins Credentials to manage resources in Azure. To create a service principle you can follow this guide if you do not have necessary permissions;

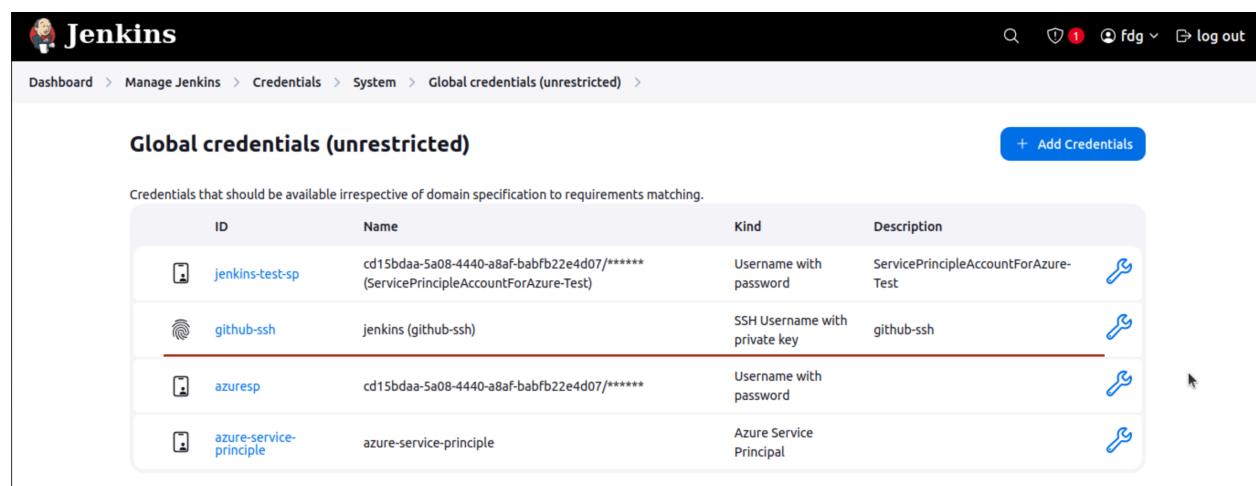
<https://learn.microsoft.com/en-us/entra/identity-platform/howto-create-service-principal-portal>

We will add the service principle to Jenkins Credentials as azure service principal. To add this as a credential on Jenkins, you need to install azure credentials plugin to Jenkins.

For more detailed steps;

<https://learn.microsoft.com/en-us/azure/developer/jenkins/deploy-to-azure-app-service-using-azure-cli>

For code checkout step from Github repo I added ssh key into Jenkins as a credential as well;



The screenshot shows the Jenkins Global credentials (unrestricted) page. The table lists the following credentials:

ID	Name	Kind	Description
jenkins-test-sp	cd15bdaa-5a08-4440-a8af-babfb22e4d07/***** (ServicePrincipleAccountForAzure-Test)	Username with password	ServicePrincipleAccountForAzure-Test
github-ssh	jenkins (github-ssh)	SSH Username with private key	github-ssh
azuresp	cd15bdaa-5a08-4440-a8af-babfb22e4d07/*****	Username with password	
azure-service-principle	azure-service-principle	Azure Service Principal	

Our pipeline will consist of 3 basic steps, 1) azure login, 2) checkout terraform codes from github 3) terraform init and apply

Here is the pipeline:

```
pipeline {  
    agent any  
    environment {  
        credentials = credentials('jenkins-test-sp')  
    }  
  
    stages {
```

```

stage('Az-login') {
    steps {
        script {
            withCredentials([azureServicePrincipal('azur...
                // Authenticate with Azure
                sh '''
                    az login --service-principal \
                    --username $AZURE_CLIENT_ID \
                    --password $AZURE_CLIENT_SECRET \
                    --tenant $AZURE_TENANT_ID
                '''
            }
        }
    }
}
stage('Checkout Code') {
    steps {
        git url: 'git@github.com:0x24dazzle/test-jenkins...
            credentialsId: 'github-ssh',
            branch: 'deploy-vm2'
    }
}
stage('Deploy') {
    steps {
        script {
            //withAzureServicePrincipal('azure-service-...
            sh '''
                export ARM_SUBSCRIPTION_ID=$AZURE_SUBSCRIPTI...
                export ARM_CLIENT_ID=$AZURE_CLIENT_ID
            '''
        }
    }
}

```

```

        export ARM_CLIENT_SECRET=$AZURE_CLIENT_SECRET
        export ARM_TENANT_ID=$AZURE_TENANT_ID

        terraform init -upgrade
        terraform init
        terraform apply -auto-approve
        ...
        //}
    }

}
}
}

```

Now let's move over to our linux terminal to create and push our terraform code.

Password authentication to github was removed. Create a public ssh and add it to your github repository. Then add your ssh repository url to .git/config file. For further information;

<https://mkyong.com/git/github-keep-asking-for-username-password-when-git-push/>

Initialize our branch

```

git branch deploy-vm2
git checkout deploy-vm2
git commit -m "first commit message"
git add .
git push -u origin deploy-vm2

```

```

nothing to commit, working tree clean
root@dazzle-virtual-machine:/home/dazzle/Documents/terraform# git branch
  deploy-vm
* deploy-vm2

```

Now we can create our terraform files;

We will use providers.tf to use azure provider;

```
terraform {
  required_providers {
    azurerm = {
      source  = "hashicorp/azurerm"
      version = "~> 3.0"
    }
  }
  required_version = ">= 1.1.3"
}
provider "azurerm" {
  features{}
}
```

Here is the variables.tf file:

```
variable "resource_group_name" {
  description = "Name of the resource group"
  default     = "test-jenkins2"
}

variable "location" {
  description = "Azure region for the resources"
  default     = "West Europe"
}

variable "admin_username" {
  description = "Admin username for the VM"
  default     = "azureuser"
}

variable "admin_password" {
  description = "Admin password for the VM"
```

```
  default      = "P@ssw0rd1234!"  
}
```

Here is the outputs.tf:

```
output "public_ip_address" {  
  description = "Public IP address of the virtual machine"  
  value       = azurerm_public_ip.public_ip.ip_address  
}
```

In [main.tf](#) we have several resources; resource group, virtual network, subnet, public\_ip, network security group (with the security rules for allowing inbound ssh connections over port22), network interface card (with the IP configuration that uses our newly created subnet and public IP explicitly), and lastly, our virtual machine with the necessary components such as storage and virtual machines image, username and password etc.

main.tf:

```
resource "azurerm_resource_group" "rg" {  
  name      = var.resource_group_name  
  location = var.location  
}  
  
resource "azurerm_virtual_network" "vnet" {  
  name          = "myVnet"  
  address_space = ["10.0.0.0/16"]  
  location      = azurerm_resource_group.rg.location  
  resource_group_name = azurerm_resource_group.rg.name  
}  
  
resource "azurerm_subnet" "subnet" {  
  name          = "mySubnet"  
  resource_group_name = azurerm_resource_group.rg.name  
  virtual_network_name = azurerm_virtual_network.vnet.name  
  address_prefixes    = ["10.0.1.0/24"]
```

```

}

resource "azurerm_public_ip" "public_ip" {
  name          = "myPublicIP"
  location      = azurerm_resource_group.rg.location
  resource_group_name = azurerm_resource_group.rg.name
  allocation_method = "Dynamic"
}

resource "azurerm_network_security_group" "nsg" {
  name          = "myNSG"
  location      = azurerm_resource_group.rg.location
  resource_group_name = azurerm_resource_group.rg.name

  security_rule {
    name          = "SSH"
    priority      = 1000
    direction     = "Inbound"
    access        = "Allow"
    protocol      = "Tcp"
    source_port_range = "*"
    destination_port_range = "22"
    source_address_prefix = "*"
    destination_address_prefix = "*"
  }
}

resource "azurerm_network_interface" "nic" {
  name          = "myNIC"
  location      = azurerm_resource_group.rg.location
  resource_group_name = azurerm_resource_group.rg.name

  ip_configuration {
    name          = "myNICConfig"
    subnet_id     = azurerm_subnet.subnet.id
    private_ip_address_allocation = "Dynamic"
  }
}

```

```

        public_ip_address_id      = azurerm_public_ip.public_ip
    }
}

resource "azurerm_linux_virtual_machine" "vm" {
    name          = "myVM"
    location      = azurerm_resource_group.rg.location
    resource_group_name = azurerm_resource_group.rg.name
    size          = "Standard_B1s"
    admin_username = var.admin_username
    admin_password = var.admin_password
    network_interface_ids = [
        azurerm_network_interface.nic.id,
    ]

    os_disk {
        caching          = "ReadWrite"
        storage_account_type = "Standard_LRS"
    }

    source_image_reference {
        publisher = "Canonical"
        offer     = "0001-com-ubuntu-server-jammy"
        sku       = "22_04-lts-gen2"
        version   = "latest"
    }
    disable_password_authentication = false
}

```

```

Processing triggers for man-db (2.10.2-1) ...
[root@dazzle-virtual-machine:/home/dazzle/Documents/terraform# vim providers.tf ] 
[root@dazzle-virtual-machine:/home/dazzle/Documents/terraform# vim variables.tf ] 
[root@dazzle-virtual-machine:/home/dazzle/Documents/terraform# vim outputs.tf ] 
[root@dazzle-virtual-machine:/home/dazzle/Documents/terraform# vim main.tf ] 
root@dazzle-virtual-machine:/home/dazzle/Documents/terraform# █

```

After adding our files we will push to our github repository;

```
git branch
git add .
git commit -m "message"
git push -u origin deploy-vm2
```

Our github looks like this;

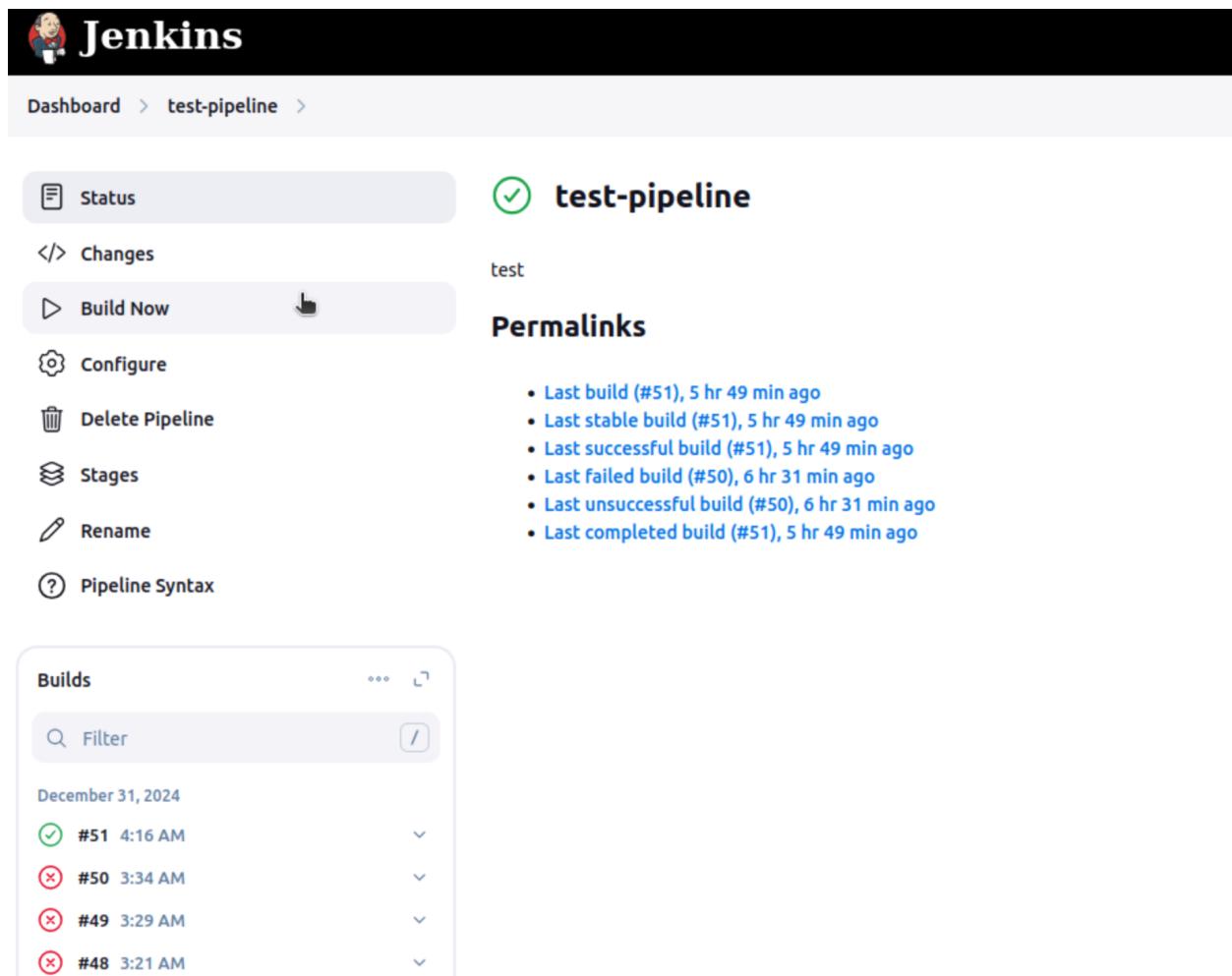
The screenshot shows a GitHub repository page for the 'test-jenkins' repository. The repository is private. A yellow banner at the top indicates that the 'deploy-vm2' branch has recent pushes 42 minutes ago. Below the banner, the repository details show 2 branches and 0 tags. A search bar and a 'Code' dropdown are visible. A message states that the branch is 11 commits ahead of 'deploy-vm'. The commit list for the 'deploy-vm2' branch shows the following commits:

Author	Message	Time	Commits
0x24dazzle	readme file changed for testing purposes	13dca9c · 42 minutes ago	19 Commits
	README.md	readme file changed for testing purposes	42 minutes ago
	main.tf	username password authentication has been enabled	5 hours ago
	outputs.tf	switched to v2	7 hours ago
	providers.tf	providers.tf missing features fixed	7 hours ago
	variables.tf	new resource group will be created	6 hours ago

## Commits

deploy-vm2	All users	All time
-o- Commits on Dec 31, 2024		
<b>readme file changed for testing purposes</b> 0x24dazzle committed 43 minutes ago		
	13dca9c	 
<b>username password authentication has been enabled</b> 0x24dazzle committed 5 hours ago		
	ec03522	 
<b>reverted ssh key change</b> 0x24dazzle committed 5 hours ago		
	3f7d076	 
<b>admin ssl certificate added</b> 0x24dazzle committed 6 hours ago		
	f06a319	 
<b>new resource group will be created</b> 0x24dazzle committed 6 hours ago		
	acf010a	 
<b>providers.tf missing features fixed</b> 0x24dazzle committed 7 hours ago		
	11446b0	 
<b>variables now do not have subs id tenant id info declaratively</b> 0x24dazzle committed 7 hours ago		
	5f3ea61	 
<b>variables inside provider have been removed</b> 0x24dazzle committed 7 hours ago		
	8d2574e	 
<b>variables file updated and tenant, subs ID are now explicitly declared</b> 0x24dazzle committed 7 hours ago		
	5e5628e	 

After building our pipeline, we can see that the resources are getting created on Azure;



The screenshot shows the Jenkins test-pipeline dashboard. The top navigation bar includes the Jenkins logo and the text "Dashboard > test-pipeline >". On the left, a sidebar lists pipeline management options: Status (highlighted in grey), Changes, Build Now (with a "Build Now" button), Configure, Delete Pipeline, Stages, Rename, and Pipeline Syntax. The main content area is titled "test-pipeline" with a green checkmark icon. It displays the text "test" and a "Permalinks" section with a list of build links. Below this is a "Builds" section with a "Builds" header, a "Filter" input field, and a table showing build history for December 31, 2024. The table includes columns for build number, status, and timestamp. Build #51 is marked as successful (green checkmark), while builds #50, #49, and #48 are marked as failed (red X).

Build	Status	Timestamp
#51	Success	4:16 AM
#50	Failure	3:34 AM
#49	Failure	3:29 AM
#48	Failure	3:21 AM

```
[0m[1mazurerm_virtual_network.vnet: Creating...[0m[0m
[0m[1mazurerm_network_security_group.nsg: Creating...[0m[0m
[0m[1mazurerm_network_security_group.nsg: Creation complete after 5s [id=/subscriptions/ba38195f-
caaf-46fd-8149-0ee5f6d5b2cc/resourceGroups/test-jenkins2/providers/Microsoft.Network/networkSecurityGroups/
myNSG][0m
[0m[1mazurerm_public_ip.public_ip: Creation complete after 5s [id=/subscriptions/ba38195f-
caaf-46fd-8149-0ee5f6d5b2cc/resourceGroups/test-jenkins2/providers/Microsoft.Network/publicIPAddresses/
myPublicIP][0m
[0m[1mazurerm_virtual_network.vnet: Creation complete after 7s [id=/subscriptions/ba38195f-
caaf-46fd-8149-0ee5f6d5b2cc/resourceGroups/test-jenkins2/providers/Microsoft.Network/virtualNetworks/myVnet][0m
[0m[1mazurerm_subnet.subnet: Creating...[0m[0m
[0m[1mazurerm_subnet.subnet: Creation complete after 8s [id=/subscriptions/ba38195f-caaf-46fd-8149-0ee5f6d5b2cc/
resourceGroups/test-jenkins2/providers/Microsoft.Network/virtualNetworks/myVnet/subnets/mySubnet][0m
[0m[1mazurerm_network_interface.nic: Creating...[0m[0m
[0m[1mazurerm_network_interface.nic: Still creating... [10s elapsed][0m[0m
[0m[1mazurerm_network_interface.nic: Creation complete after 14s [id=/subscriptions/ba38195f-
caaf-46fd-8149-0ee5f6d5b2cc/resourceGroups/test-jenkins2/providers/Microsoft.Network/networkInterfaces/myNIC][0m
[0m[1mazurerm_linux_virtual_machine.vm: Creating...[0m[0m
[0m[1mazurerm_linux_virtual_machine.vm: Still creating... [10s elapsed][0m[0m
[0m[1mazurerm_linux_virtual_machine.vm: Still creating... [20s elapsed][0m[0m
[0m[1mazurerm_linux_virtual_machine.vm: Still creating... [30s elapsed][0m[0m
[0m[1mazurerm_linux_virtual_machine.vm: Still creating... [40s elapsed][0m[0m
[0m[1mazurerm_linux_virtual_machine.vm: Still creating... [50s elapsed][0m[0m
[0m[1mazurerm_linux_virtual_machine.vm: Creation complete after 53s [id=/subscriptions/ba38195f-
caaf-46fd-8149-0ee5f6d5b2cc/resourceGroups/test-jenkins2/providers/Microsoft.Compute/virtualMachines/myVM][0m
[0m[1m[32m
Apply complete! Resources: 7 added, 0 changed, 0 destroyed.
[0m[0m[1m[32m
```

Resource groups		
Default Directory (0x24ph@gmail.onmicrosoft.com)		
<a href="#">+ Create</a> <a href="#">Manage view</a> <a href="#">Refresh</a> <a href="#">Export to CSV</a> <a href="#">Open query</a> <a href="#">Assign tags</a>		
<input type="checkbox"/> <a href="#">Filter for any field...</a>	<a href="#">Subscription equals all</a>	<a href="#">Location equals all</a> <a href="#">X</a> <a href="#">Add filter</a>
Showing 1 to 3 of 3 records.		<a href="#">No grouping</a> <a href="#">List view</a>
<input type="checkbox"/> <a href="#">Name ↑↓</a>	<a href="#">Subscription ↑↓</a>	<a href="#">Location ↑↓</a>
<input type="checkbox"/> <a href="#">[NetworkWatcherRG]</a>	<a href="#">Azure subscription 1</a>	<a href="#">West Europe</a> <a href="#">...</a>
<input type="checkbox"/> <a href="#">[test-jenkins]</a>	<a href="#">Azure subscription 1</a>	<a href="#">West Europe</a> <a href="#">...</a>
<input type="checkbox"/> <a href="#">[test-jenkins2]</a>	<a href="#">Azure subscription 1</a>	<a href="#">West Europe</a> <a href="#">...</a>

Home > Resource groups > **test-jenkins2** ...

**Overview**

**Essentials**

**Resources** Recommendations (2)

Showing 1 to 6 of 6 records.  Show hidden types ...

<input type="checkbox"/> Name ↑	Type ↑	Location ↑
<input type="checkbox"/> myNIC	Network Interface	West Europe
<input type="checkbox"/> myNSG	Network security group	West Europe
<input type="checkbox"/> myPublicIP	Public IP address	West Europe
<input type="checkbox"/> myVM	Virtual machine	West Europe
<input type="checkbox"/> myVM_OsDisk_1	Disk	West Europe
<input type="checkbox"/> myVnet	Virtual network	West Europe

Home > Resource groups > **test-jenkins2** > **myVM** ...

**Overview**

**Help me copy this VM in any region**

**Connect** ... **Start** ... **Restart** ... **Stop** ... **Hibernate** ... **Capture** ... **Delete** ... **Refresh** ... **Open in mobile** ... **Feedback** ...

**Essentials**

Resource group ( <a href="#">move</a> ) <a href="#">test-jenkins2</a>	Operating system Linux (ubuntu 22.04)
Status Running	Size Standard B1s (1 vcpu, 1 GiB memory)
Location West Europe	Public IP address <a href="#">23.97.188.246</a>
Subscription ( <a href="#">move</a> ) <a href="#">Azure subscription 1</a>	Virtual network/subnet <a href="#">myVnet/mySubnet</a>
Subscription ID <a href="#">ba38195f-caaf-46fd-8149-0ee5f6d5b2cc</a>	DNS name <a href="#">Not configured</a>
	Health state -
	Time created 12/31/2024, 4:17 AM UTC

**Tags** [edit](#) [Add tags](#)

Let's connect to our VM using public IP address and username password;

myVM | Connect

Virtual machine

Search Refresh Troubleshoot More Options Feedback

Overview Activity log Access control (IAM) Tags Diagnose and solve problems Connect Connect

Connecting using Public IP address | 23.97.188.246

Admin username: azureuser  
Port (change): 22 Check access

Just-in-time policy: Unsupported by plan

```

root@dazzle-virtual-machine:/home/dazzle/Documents/terraform# ssh azureuser@23.97.188.246
azureuser@23.97.188.246's password:
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 6.5.0-1025-azure x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

System information as of Tue Dec 31 10:10:52 UTC 2024

  System load:  0.03           Processes:          105
  Usage of /:   7.1% of 28.89GB  Users logged in:   0
  Memory usage: 35%           IPv4 address for eth0: 10.0.1.4
  Swap usage:   0%

* Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
  just raised the bar for easy, resilient and secure K8s cluster deployment.

  https://ubuntu.com/engage/secure-kubernetes-at-the-edge

Expanded Security Maintenance for Applications is not enabled.

12 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

New release '24.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

*** System restart required ***
Last login: Tue Dec 31 09:23:37 2024 from 85.22.68.226
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

azureuser@myVM:~$
```

What could be improved further;

1. Ssh keys could be added to enhance security
2. Instead of using public IP address, a bastion or a load balancer could be used

3. Tags can be used to improve resource organization and cost tracking
4. depends\_on can be used to be sure to implement resources in the right order
5. Diagnostic tools can be used to troubleshoot any issues