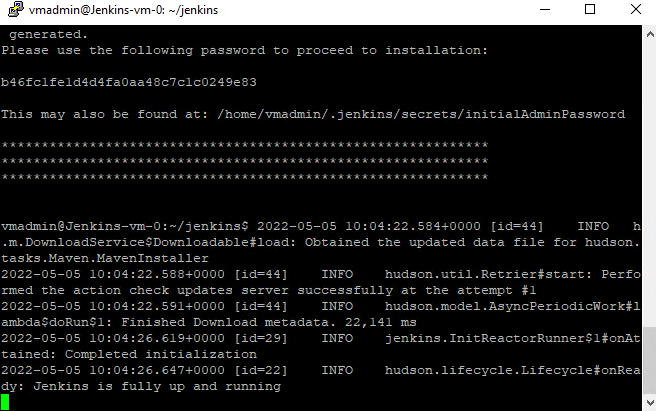
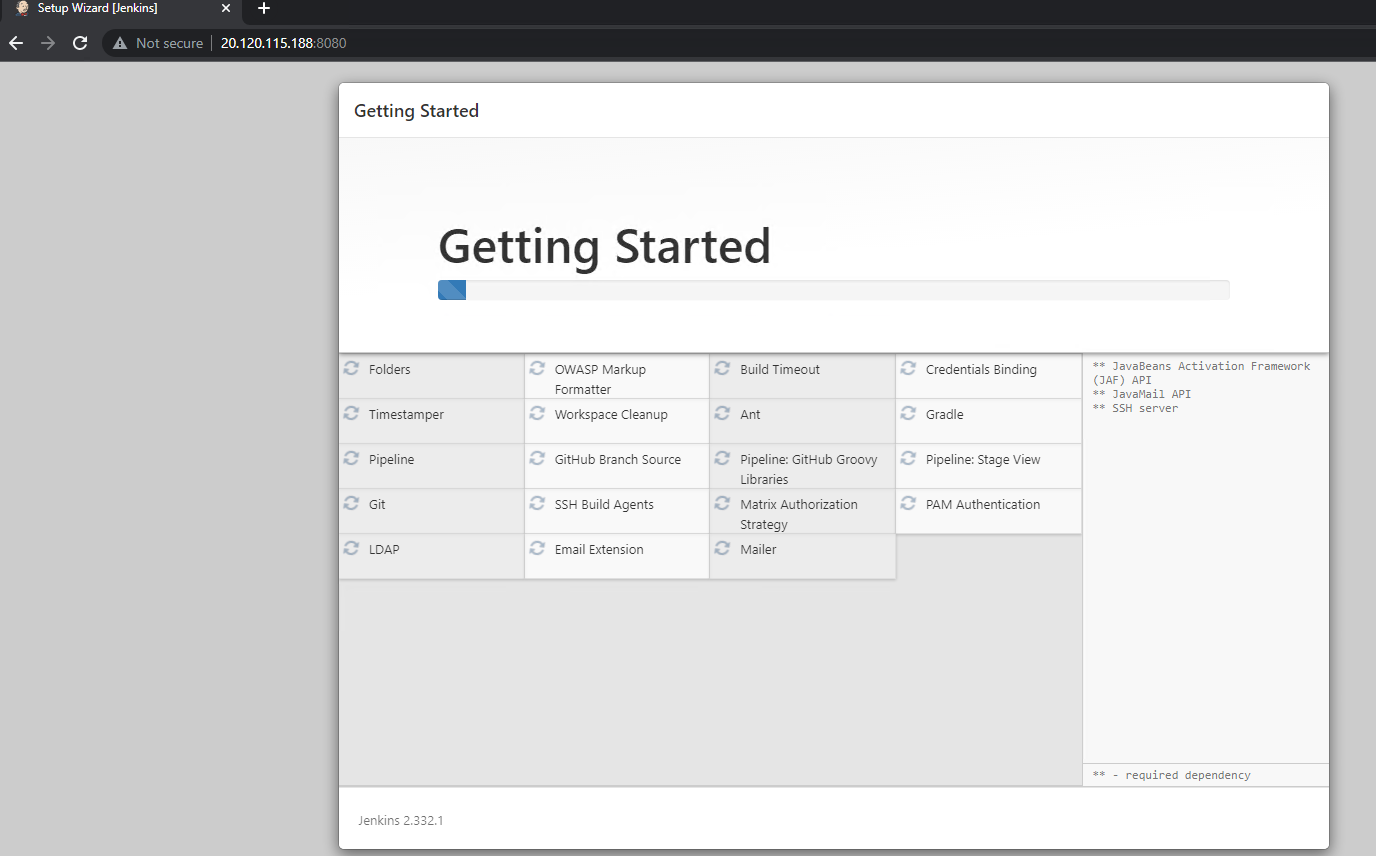
|  |  |
| --- | --- |
| **Task** | **Technology** |
| **Part 1** | |
| Use Azurerm resource provider | Terraform |
| Generate Service Principle for Terraform in Azure | Azure |
| Create required Vars.tf to create two VMs in Azure | Terraform |
| Create main.tf to create two Linux VMs | Terraform |
| Use Terraform Provisioner to install JDK and Jenkins in VM1 | Terraform |
| Use Terraform Provisioner to install JDK, Maven,Ansible,Docker,AzureCli and Git | Terraform |
| Init,Plan and Apply Terraform Script | Terraform |
| Manually Start Jenkins and configure required Plug-ins and Master Slave Configuration | Jenkins |
| **Part 2 - Phase1** | |
| Create Maven Project with Archtype as web application in eclipse | Eclipse |
| Modify Index.jsp under src/main/webcontent to display a custom message | Eclipse |
| Run Maven clean install in eclipse to check the build and ckeck for .war file in target folder | Eclipse |
| **Part 2- Phase 2** | |
| Gerate Dockerfile under project folder of your app | Eclipse |
| Modify FORM statement to use tomcat as base image | Eclipse |
| Test the Dockerfile by running Docker build and create a container | Docker |
| Access the application from container and check it | Browser |
| **Part 2-** | |
| Create a github repository and copy repo URL | Github Site |
| In Eclipse convert the app in to a local repo from Team meanu share Project Option | Eclipse |
| Commit and Push the code to remote repo | Eclipse |
| **Part 2- Phase 4** | |
| Modify Project index.jsp, Commit and Push to remote repo | Eclipse |
| Check for the change in remote Repo | Github Site |
| **Part 2- Phase 5** | |
| In build server configure Ansible manually | Shell |
| Modify ansible.cfg to use hosts file as inventory | Shell |
| install python-pip in ansible server | apt-get |
| Using PIP install azure modules in Ansible server | PIP |
| Use the same service Principle created for terraform for ansible to get authenticated to Azure | Azure Portal |
| Create a playbook1 to create aks cluster in azure and test | vi |
| **Part 2- Phase 6** | |
| Push playbook1 to Remote Git repo created | Git Bash |
| Check for the change in remote Repo | Github Site |
| **Part 2- Phase 7** | |
| From Build server run the playbook1 to test for the required result and also create yamls to deploy image to AKS cluster and rollback. | Ansible and Shell |
| Push Manifest YAMLs to the Git Repo | Git Bash |
| **Part 2- Phase 8** | |
| Create Repo in hub.docker.com or acr | Docker hub Site |
| Configure Global tool configurations in Jenkins to use JDK,Maven and Git | Jenkins Web UI |
| Configure Git credentials in Jenkins Vault | Jenkins Web UI |
| Create Pipeline1 using Freestyle project in Jenkins | Jenkins Web UI |
| In SCM stage Pull code form Remote Repo | Jenkins Web UI |
| In Build Stage, Step1 : use maven top level target to build | Jenkins Web UI |
| In Build Stage Step 2: User Docker build and Push to create image whih contains your app and push to Docker Hub or acr | Jenkins Web UI |
| Create release pipeline in Azure Devops Project | Azure Devops |
| In Artifacts pull plybook and k8s application manifests (YAMLs) form Remote Repo | Azure Devops |
| In Build Stage Step 1: Run ansible Playbook1 | Azure Devops |
| In Build Stage Step2 : Deploy the pods to AKS Cluster using image created in Docker HUB or ACR | Azure Devops |
| In Github repo configure webhook for Jenkins | Github Site |
| For jenkins Pipeline configure Build trigger to use Github webhook for continuous integration | Jenkins Web UI |

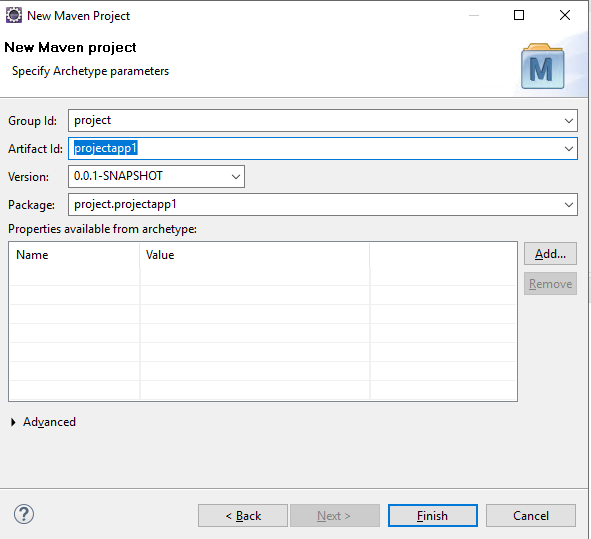
|  |
| --- |
| provider "azurerm" {      features {}      subscription\_id = "6ca1dc7b-a8e2--0603f8007f95"      client\_id       = "2af2b703-f145-9a23-3dd81522282f"      client\_secret   = "xjS7Q~Ql8uGPdJBCan0\_.-XNnru8"      tenant\_id       = "d0920dec-0ac7-bd47-e3cdb559d9b4"  }  variable "vm\_name\_pfx" {    description = "VM Names"    default     = "Jenkins-vm"    type        = string  }  variable "vm\_count" {    description = "Number of Virtual Machines"    default     = 2    type        = string  }  resource "azurerm\_resource\_group" "main" {    name     = "Project-Rg1"    location = "eastus"  }  resource "azurerm\_virtual\_network" "main" {    name                = "vnet1-network"    address\_space       = ["10.0.0.0/16"]    location            = "eastus"    resource\_group\_name = azurerm\_resource\_group.main.name  }  resource "azurerm\_subnet" "main" {    name                 = "internal"    resource\_group\_name  = azurerm\_resource\_group.main.name    virtual\_network\_name = azurerm\_virtual\_network.main.name    address\_prefixes     = ["10.0.2.0/24"]  }  resource "azurerm\_public\_ip" "main" {  count               = var.vm\_count    name                = "${var.vm\_name\_pfx}-${count.index}-pip"    resource\_group\_name = azurerm\_resource\_group.main.name    location            = "eastus"    allocation\_method   = "Dynamic"  }  resource "azurerm\_network\_interface" "main" {    count               = var.vm\_count    name                = "${var.vm\_name\_pfx}-${count.index}-nic"    location            = "eastus"    resource\_group\_name = azurerm\_resource\_group.main.name      ip\_configuration {      name                          = "internal"      subnet\_id                     = azurerm\_subnet.main.id      private\_ip\_address\_allocation = "Dynamic"      public\_ip\_address\_id          = azurerm\_public\_ip.main[count.index].id    }  }  resource "azurerm\_network\_security\_group" "main" {    name                = "${var.vm\_name\_pfx}-nsg"    location            = azurerm\_resource\_group.main.location    resource\_group\_name = azurerm\_resource\_group.main.name    security\_rule {      name                       = "${var.vm\_name\_pfx}-rule"      priority                   = 100      direction                  = "Inbound"      access                     = "Allow"      protocol                   = "Tcp"      source\_port\_range          = "\*"      destination\_port\_range     = "\*"      source\_address\_prefix      = "\*"      destination\_address\_prefix = "\*"    }  }  resource "azurerm\_subnet\_network\_security\_group\_association" "main" {    subnet\_id                 = azurerm\_subnet.main.id    network\_security\_group\_id = azurerm\_network\_security\_group.main.id  }  resource "azurerm\_linux\_virtual\_machine" "main" {    count               = var.vm\_count # Count Value read from variable    name                = "${var.vm\_name\_pfx}-${count.index}" # Name constructed using count and pfx    resource\_group\_name = azurerm\_resource\_group.main.name    location            = "eastus"    size                = "Standard\_B1s"    admin\_username      = "vmadmin"    admin\_password      = "p@ssw0rd13579"    disable\_password\_authentication = false    network\_interface\_ids = [      azurerm\_network\_interface.main[count.index].id,    ]    os\_disk {      storage\_account\_type = "Standard\_LRS"      caching              = "ReadWrite"    }     source\_image\_reference {      publisher = "Canonical"      offer     = "UbuntuServer"      sku       = "18.04-LTS"      version   = "latest"    }  }  output "public\_ip\_address" {    value = "${azurerm\_public\_ip.main.\*.ip\_address}"  }  resource "null\_resource" "nr1" {    provisioner "remote-exec" {      inline = [        "sudo apt-get update && sudo apt install -y openjdk-11-jre-headless",        "mkdir jenkins && cd jenkins",        "wget https://get.jenkins.io/war-stable/2.332.1/jenkins.war"      ]      connection {        host     = "${element(azurerm\_linux\_virtual\_machine.main.\*.public\_ip\_address, 0)}"        user     = "vmadmin"        password = "p@ssw0rd13579"      }  }  }  resource "null\_resource" "nr2" {    provisioner "remote-exec" {      inline = [        "sudo apt-get update && sudo apt install -y openjdk-11-jre-headless",        "sudo apt-get install -y maven",        "sudo apt-get install -y docker\*",        "sudo apt-get install software-properties-common && sudo apt-add-repository ppa:ansible/ansible -y && sudo apt-get update && sudo apt-get install -y ansible",        "sudo chown vmadmin:vmadmin /etc/ansible/ && sudo usermod -aG docker $USER && sudo chmod 666 /var/run/docker.sock && sudo apt install -y gnupg2 pass",        "sudo apt-get update && sudo apt-get install -y apt-transport-https gnupg2 curl",        "curl -sL https://aka.ms/InstallAzureCLIDeb | sudo bash",        "sudo apt update && sudo apt install -y git"      ]      connection {        host     = "${element(azurerm\_linux\_virtual\_machine.main.\*.public\_ip\_address, 1)}"        user     = "vmadmin"        password = "p@ssw0rd13579"      }  }  } |

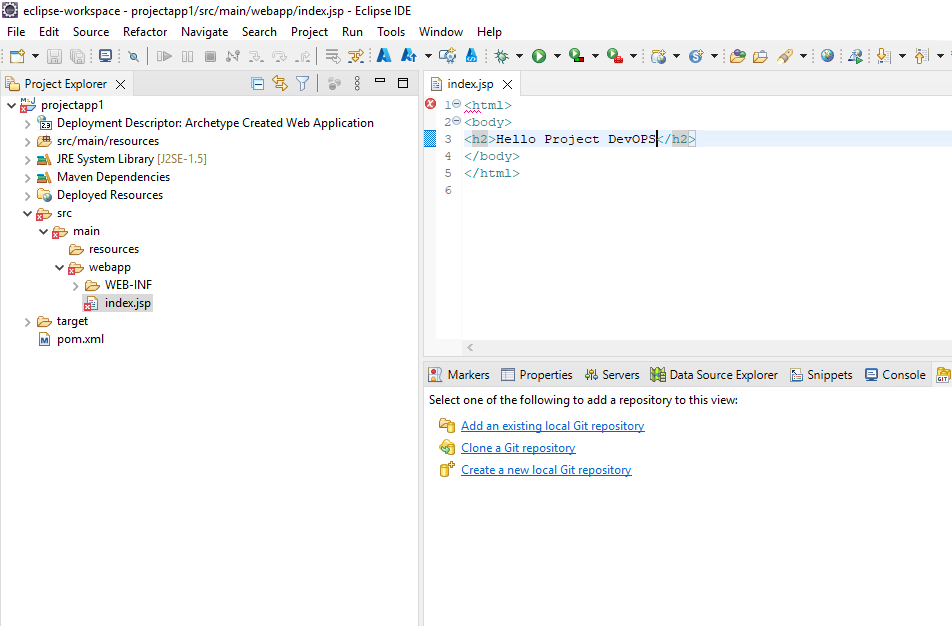
|  |
| --- |
| PS C:\tfdemos\project> terraform init  Initializing the backend...  Initializing provider plugins...  - Finding latest version of hashicorp/azurerm...  - Finding latest version of hashicorp/null...  - Installing hashicorp/azurerm v3.4.0...  - Installed hashicorp/azurerm v3.4.0 (signed by HashiCorp)  - Installing hashicorp/null v3.1.1...  - Installed hashicorp/null v3.1.1 (signed by HashiCorp)  Terraform has created a lock file .terraform.lock.hcl to record the provider  selections it made above. Include this file in your version control repository  so that Terraform can guarantee to make the same selections by default when  you run "terraform init" in the future.  Terraform has been successfully initialized!  You may now begin working with Terraform. Try running "terraform plan" to see  any changes that are required for your infrastructure. All Terraform commands  should now work.  If you ever set or change modules or backend configuration for Terraform,  rerun this command to reinitialize your working directory. If you forget, other  commands will detect it and remind you to do so if necessary.  PS C:\tfdemos\project> terraform plan  Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:  + create  Terraform will perform the following actions:  # azurerm\_linux\_virtual\_machine.main[0] will be created  + resource "azurerm\_linux\_virtual\_machine" "main" {  + admin\_password = (sensitive value)  + admin\_username = "vmadmin"  + allow\_extension\_operations = true  + computer\_name = (known after apply)  + disable\_password\_authentication = false  + extensions\_time\_budget = "PT1H30M"  + id = (known after apply)  + location = "eastus"  + max\_bid\_price = -1  + name = "Jenkins-vm-0"  + network\_interface\_ids = (known after apply)  + patch\_mode = "ImageDefault"  + platform\_fault\_domain = -1  + priority = "Regular"  + private\_ip\_address = (known after apply)  + private\_ip\_addresses = (known after apply)  + provision\_vm\_agent = true  + public\_ip\_address = (known after apply)  + public\_ip\_addresses = (known after apply)  + resource\_group\_name = "Project-Rg1"  + size = "Standard\_B1s"  + virtual\_machine\_id = (known after apply)  + os\_disk {  + caching = "ReadWrite"  + disk\_size\_gb = (known after apply)  + name = (known after apply)  + storage\_account\_type = "Standard\_LRS"  + write\_accelerator\_enabled = false  }  + source\_image\_reference {  + offer = "UbuntuServer"  + publisher = "Canonical"  + sku = "18.04-LTS"  + version = "latest"  }  + termination\_notification {  + enabled = (known after apply)  + timeout = (known after apply)  }  }  # azurerm\_linux\_virtual\_machine.main[1] will be created  + resource "azurerm\_linux\_virtual\_machine" "main" {  + admin\_password = (sensitive value)  + admin\_username = "vmadmin"  + allow\_extension\_operations = true  + computer\_name = (known after apply)  + disable\_password\_authentication = false  + extensions\_time\_budget = "PT1H30M"  + id = (known after apply)  + location = "eastus"  + max\_bid\_price = -1  + name = "Jenkins-vm-1"  + network\_interface\_ids = (known after apply)  + patch\_mode = "ImageDefault"  + platform\_fault\_domain = -1  + priority = "Regular"  + private\_ip\_address = (known after apply)  + private\_ip\_addresses = (known after apply)  + provision\_vm\_agent = true  + public\_ip\_address = (known after apply)  + public\_ip\_addresses = (known after apply)  + resource\_group\_name = "Project-Rg1"  + size = "Standard\_B1s"  + virtual\_machine\_id = (known after apply)  + os\_disk {  + caching = "ReadWrite"  + disk\_size\_gb = (known after apply)  + name = (known after apply)  + storage\_account\_type = "Standard\_LRS"  + write\_accelerator\_enabled = false  }  + source\_image\_reference {  + offer = "UbuntuServer"  + publisher = "Canonical"  + sku = "18.04-LTS"  + version = "latest"  }  + termination\_notification {  + enabled = (known after apply)  + timeout = (known after apply)  }  }  # azurerm\_network\_interface.main[0] will be created  + resource "azurerm\_network\_interface" "main" {  + applied\_dns\_servers = (known after apply)  + dns\_servers = (known after apply)  + enable\_accelerated\_networking = false  + enable\_ip\_forwarding = false  + id = (known after apply)  + internal\_dns\_name\_label = (known after apply)  + internal\_domain\_name\_suffix = (known after apply)  + location = "eastus"  + mac\_address = (known after apply)  + name = "Jenkins-vm-0-nic"  + private\_ip\_address = (known after apply)  + private\_ip\_addresses = (known after apply)  + resource\_group\_name = "Project-Rg1"  + virtual\_machine\_id = (known after apply)  + ip\_configuration {  + gateway\_load\_balancer\_frontend\_ip\_configuration\_id = (known after apply)  + name = "internal"  + primary = (known after apply)  + private\_ip\_address = (known after apply)  + private\_ip\_address\_allocation = "Dynamic"  + private\_ip\_address\_version = "IPv4"  + public\_ip\_address\_id = (known after apply)  + subnet\_id = (known after apply)  }  }  # azurerm\_network\_interface.main[1] will be created  + resource "azurerm\_network\_interface" "main" {  + applied\_dns\_servers = (known after apply)  + dns\_servers = (known after apply)  + enable\_accelerated\_networking = false  + enable\_ip\_forwarding = false  + id = (known after apply)  + internal\_dns\_name\_label = (known after apply)  + internal\_domain\_name\_suffix = (known after apply)  + location = "eastus"  + mac\_address = (known after apply)  + name = "Jenkins-vm-1-nic"  + private\_ip\_address = (known after apply)  + private\_ip\_addresses = (known after apply)  + resource\_group\_name = "Project-Rg1"  + virtual\_machine\_id = (known after apply)  + ip\_configuration {  + gateway\_load\_balancer\_frontend\_ip\_configuration\_id = (known after apply)  + name = "internal"  + primary = (known after apply)  + private\_ip\_address = (known after apply)  + private\_ip\_address\_allocation = "Dynamic"  + private\_ip\_address\_version = "IPv4"  + public\_ip\_address\_id = (known after apply)  + subnet\_id = (known after apply)  }  }  # azurerm\_network\_security\_group.main will be created  + resource "azurerm\_network\_security\_group" "main" {  + id = (known after apply)  + location = "eastus"  + name = "Jenkins-vm-nsg"  + resource\_group\_name = "Project-Rg1"  + security\_rule = [  + {  + access = "Allow"  + description = ""  + destination\_address\_prefix = "\*"  + destination\_address\_prefixes = []  + destination\_application\_security\_group\_ids = []  + destination\_port\_range = "\*"  + destination\_port\_ranges = []  + direction = "Inbound"  + name = "Jenkins-vm-rule"  + priority = 100  + protocol = "Tcp"  + source\_address\_prefix = "\*"  + source\_address\_prefixes = []  + source\_application\_security\_group\_ids = []  + source\_port\_range = "\*"  + source\_port\_ranges = []  },  ]  }  # azurerm\_public\_ip.main[0] will be created  + resource "azurerm\_public\_ip" "main" {  + allocation\_method = "Dynamic"  + fqdn = (known after apply)  + id = (known after apply)  + idle\_timeout\_in\_minutes = 4  + ip\_address = (known after apply)  + ip\_version = "IPv4"  + location = "eastus"  + name = "Jenkins-vm-0-pip"  + resource\_group\_name = "Project-Rg1"  + sku = "Basic"  + sku\_tier = "Regional"  }  # azurerm\_public\_ip.main[1] will be created  + resource "azurerm\_public\_ip" "main" {  + allocation\_method = "Dynamic"  + fqdn = (known after apply)  + id = (known after apply)  + idle\_timeout\_in\_minutes = 4  + ip\_address = (known after apply)  + ip\_version = "IPv4"  + location = "eastus"  + name = "Jenkins-vm-1-pip"  + resource\_group\_name = "Project-Rg1"  + sku = "Basic"  + sku\_tier = "Regional"  }  # azurerm\_resource\_group.main will be created  + resource "azurerm\_resource\_group" "main" {  + id = (known after apply)  + location = "eastus"  + name = "Project-Rg1"  }  # azurerm\_subnet.main will be created  + resource "azurerm\_subnet" "main" {  + address\_prefixes = [  + "10.0.2.0/24",  ]  + enforce\_private\_link\_endpoint\_network\_policies = false  + enforce\_private\_link\_service\_network\_policies = false  + id = (known after apply)  + name = "internal"  + resource\_group\_name = "Project-Rg1"  + virtual\_network\_name = "vnet1-network"  }  # azurerm\_subnet\_network\_security\_group\_association.main will be created  + resource "azurerm\_subnet\_network\_security\_group\_association" "main" {  + id = (known after apply)  + network\_security\_group\_id = (known after apply)  + subnet\_id = (known after apply)  }  # azurerm\_virtual\_network.main will be created  + resource "azurerm\_virtual\_network" "main" {  + address\_space = [  + "10.0.0.0/16",  ]  + dns\_servers = (known after apply)  + guid = (known after apply)  + id = (known after apply)  + location = "eastus"  + name = "vnet1-network"  + resource\_group\_name = "Project-Rg1"  + subnet = (known after apply)  }  # null\_resource.nr1 will be created  + resource "null\_resource" "nr1" {  + id = (known after apply)  }  # null\_resource.nr2 will be created  + resource "null\_resource" "nr2" {  + id = (known after apply)  }  Plan: 13 to add, 0 to change, 0 to destroy.  Changes to Outputs:  + public\_ip\_address = [  + (known after apply),  + (known after apply),  ]  ────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────  Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run "terraform apply" now.  PS C:\tfdemos\project> terraform apply  Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:  + create  Terraform will perform the following actions:  # azurerm\_linux\_virtual\_machine.main[0] will be created  + resource "azurerm\_linux\_virtual\_machine" "main" {  + admin\_password = (sensitive value)  + admin\_username = "vmadmin"  + allow\_extension\_operations = true  + computer\_name = (known after apply)  + disable\_password\_authentication = false  + extensions\_time\_budget = "PT1H30M"  + id = (known after apply)  + location = "eastus"  + max\_bid\_price = -1  + name = "Jenkins-vm-0"  + network\_interface\_ids = (known after apply)  + patch\_mode = "ImageDefault"  + platform\_fault\_domain = -1  + priority = "Regular"  + private\_ip\_address = (known after apply)  + private\_ip\_addresses = (known after apply)  + provision\_vm\_agent = true  + public\_ip\_address = (known after apply)  + public\_ip\_addresses = (known after apply)  + resource\_group\_name = "Project-Rg1"  + size = "Standard\_B1s"  + virtual\_machine\_id = (known after apply)  + os\_disk {  + caching = "ReadWrite"  + disk\_size\_gb = (known after apply)  + name = (known after apply)  + storage\_account\_type = "Standard\_LRS"  + write\_accelerator\_enabled = false  }  + source\_image\_reference {  + offer = "UbuntuServer"  + publisher = "Canonical"  + sku = "18.04-LTS"  + version = "latest"  }  + termination\_notification {  + enabled = (known after apply)  + timeout = (known after apply)  }  }  # azurerm\_linux\_virtual\_machine.main[1] will be created  + resource "azurerm\_linux\_virtual\_machine" "main" {  + admin\_password = (sensitive value)  + admin\_username = "vmadmin"  + allow\_extension\_operations = true  + computer\_name = (known after apply)  + disable\_password\_authentication = false  + extensions\_time\_budget = "PT1H30M"  + id = (known after apply)  + location = "eastus"  + max\_bid\_price = -1  + name = "Jenkins-vm-1"  + network\_interface\_ids = (known after apply)  + patch\_mode = "ImageDefault"  + platform\_fault\_domain = -1  + priority = "Regular"  + private\_ip\_address = (known after apply)  + private\_ip\_addresses = (known after apply)  + provision\_vm\_agent = true  + public\_ip\_address = (known after apply)  + public\_ip\_addresses = (known after apply)  + resource\_group\_name = "Project-Rg1"  + size = "Standard\_B1s"  + virtual\_machine\_id = (known after apply)  + os\_disk {  + caching = "ReadWrite"  + disk\_size\_gb = (known after apply)  + name = (known after apply)  + storage\_account\_type = "Standard\_LRS"  + write\_accelerator\_enabled = false  }  + source\_image\_reference {  + offer = "UbuntuServer"  + publisher = "Canonical"  + sku = "18.04-LTS"  + version = "latest"  }  + termination\_notification {  + enabled = (known after apply)  + timeout = (known after apply)  }  }  # azurerm\_network\_interface.main[0] will be created  + resource "azurerm\_network\_interface" "main" {  + applied\_dns\_servers = (known after apply)  + dns\_servers = (known after apply)  + enable\_accelerated\_networking = false  + enable\_ip\_forwarding = false  + id = (known after apply)  + internal\_dns\_name\_label = (known after apply)  + internal\_domain\_name\_suffix = (known after apply)  + location = "eastus"  + mac\_address = (known after apply)  + name = "Jenkins-vm-0-nic"  + private\_ip\_address = (known after apply)  + private\_ip\_addresses = (known after apply)  + resource\_group\_name = "Project-Rg1"  + virtual\_machine\_id = (known after apply)  + ip\_configuration {  + gateway\_load\_balancer\_frontend\_ip\_configuration\_id = (known after apply)  + name = "internal"  + primary = (known after apply)  + private\_ip\_address = (known after apply)  + private\_ip\_address\_allocation = "Dynamic"  + private\_ip\_address\_version = "IPv4"  + public\_ip\_address\_id = (known after apply)  + subnet\_id = (known after apply)  }  }  # azurerm\_network\_interface.main[1] will be created  + resource "azurerm\_network\_interface" "main" {  + applied\_dns\_servers = (known after apply)  + dns\_servers = (known after apply)  + enable\_accelerated\_networking = false  + enable\_ip\_forwarding = false  + id = (known after apply)  + internal\_dns\_name\_label = (known after apply)  + internal\_domain\_name\_suffix = (known after apply)  + location = "eastus"  + mac\_address = (known after apply)  + name = "Jenkins-vm-1-nic"  + private\_ip\_address = (known after apply)  + private\_ip\_addresses = (known after apply)  + resource\_group\_name = "Project-Rg1"  + virtual\_machine\_id = (known after apply)  + ip\_configuration {  + gateway\_load\_balancer\_frontend\_ip\_configuration\_id = (known after apply)  + name = "internal"  + primary = (known after apply)  + private\_ip\_address = (known after apply)  + private\_ip\_address\_allocation = "Dynamic"  + private\_ip\_address\_version = "IPv4"  + public\_ip\_address\_id = (known after apply)  + subnet\_id = (known after apply)  }  }  # azurerm\_network\_security\_group.main will be created  + resource "azurerm\_network\_security\_group" "main" {  + id = (known after apply)  + location = "eastus"  + name = "Jenkins-vm-nsg"  + resource\_group\_name = "Project-Rg1"  + security\_rule = [  + {  + access = "Allow"  + description = ""  + destination\_address\_prefix = "\*"  + destination\_address\_prefixes = []  + destination\_application\_security\_group\_ids = []  + destination\_port\_range = "\*"  + destination\_port\_ranges = []  + direction = "Inbound"  + name = "Jenkins-vm-rule"  + priority = 100  + protocol = "Tcp"  + source\_address\_prefix = "\*"  + source\_address\_prefixes = []  + source\_application\_security\_group\_ids = []  + source\_port\_range = "\*"  + source\_port\_ranges = []  },  ]  }  # azurerm\_public\_ip.main[0] will be created  + resource "azurerm\_public\_ip" "main" {  + allocation\_method = "Dynamic"  + fqdn = (known after apply)  + id = (known after apply)  + idle\_timeout\_in\_minutes = 4  + ip\_address = (known after apply)  + ip\_version = "IPv4"  + location = "eastus"  + name = "Jenkins-vm-0-pip"  + resource\_group\_name = "Project-Rg1"  + sku = "Basic"  + sku\_tier = "Regional"  }  # azurerm\_public\_ip.main[1] will be created  + resource "azurerm\_public\_ip" "main" {  + allocation\_method = "Dynamic"  + fqdn = (known after apply)  + id = (known after apply)  + idle\_timeout\_in\_minutes = 4  + ip\_address = (known after apply)  + ip\_version = "IPv4"  + location = "eastus"  + name = "Jenkins-vm-1-pip"  + resource\_group\_name = "Project-Rg1"  + sku = "Basic"  + sku\_tier = "Regional"  }  # azurerm\_resource\_group.main will be created  + resource "azurerm\_resource\_group" "main" {  + id = (known after apply)  + location = "eastus"  + name = "Project-Rg1"  }  # azurerm\_subnet.main will be created  + resource "azurerm\_subnet" "main" {  + address\_prefixes = [  + "10.0.2.0/24",  ]  + enforce\_private\_link\_endpoint\_network\_policies = false  + enforce\_private\_link\_service\_network\_policies = false  + id = (known after apply)  + name = "internal"  + resource\_group\_name = "Project-Rg1"  + virtual\_network\_name = "vnet1-network"  }  # azurerm\_subnet\_network\_security\_group\_association.main will be created  + resource "azurerm\_subnet\_network\_security\_group\_association" "main" {  + id = (known after apply)  + network\_security\_group\_id = (known after apply)  + subnet\_id = (known after apply)  }  # azurerm\_virtual\_network.main will be created  + resource "azurerm\_virtual\_network" "main" {  + address\_space = [  + "10.0.0.0/16",  ]  + dns\_servers = (known after apply)  + guid = (known after apply)  + id = (known after apply)  + location = "eastus"  + name = "vnet1-network"  + resource\_group\_name = "Project-Rg1"  + subnet = (known after apply)  }  # null\_resource.nr1 will be created  + resource "null\_resource" "nr1" {  + id = (known after apply)  }  # null\_resource.nr2 will be created  + resource "null\_resource" "nr2" {  + id = (known after apply)  }  Plan: 13 to add, 0 to change, 0 to destroy.  Changes to Outputs:  + public\_ip\_address = [  + (known after apply),  + (known after apply),  ]  Do you want to perform these actions?  Terraform will perform the actions described above.  Only 'yes' will be accepted to approve.  Enter a value: yes  azurerm\_resource\_group.main: Creating...  azurerm\_resource\_group.main: Creation complete after 0s [id=/subscriptions/6ca1dc7b-a8e2-47be-b9ef-0603f8007f95/resourceGroups/Project-Rg1]  azurerm\_virtual\_network.main: Creating...  azurerm\_public\_ip.main[0]: Creating...  azurerm\_network\_security\_group.main: Creating...  azurerm\_public\_ip.main[1]: Creating...  azurerm\_public\_ip.main[0]: Creation complete after 2s [id=/subscriptions/6ca1dc7b-a8e2-47be-b9ef-0603f8007f95/resourceGroups/Project-Rg1/providers/Microsoft.Network/publicIPAddresses/Jenkins-vm-0-pip]  azurerm\_public\_ip.main[1]: Creation complete after 2s [id=/subscriptions/6ca1dc7b-a8e2-47be-b9ef-0603f8007f95/resourceGroups/Project-Rg1/providers/Microsoft.Network/publicIPAddresses/Jenkins-vm-1-pip]  azurerm\_virtual\_network.main: Creation complete after 4s [id=/subscriptions/6ca1dc7b-a8e2-47be-b9ef-0603f8007f95/resourceGroups/Project-Rg1/providers/Microsoft.Network/virtualNetworks/vnet1-network]  azurerm\_subnet.main: Creating...  azurerm\_network\_security\_group.main: Creation complete after 4s [id=/subscriptions/6ca1dc7b-a8e2-47be-b9ef-0603f8007f95/resourceGroups/Project-Rg1/providers/Microsoft.Network/networkSecurityGroups/Jenkins-vm-nsg]  azurerm\_subnet.main: Creation complete after 4s [id=/subscriptions/6ca1dc7b-a8e2-47be-b9ef-0603f8007f95/resourceGroups/Project-Rg1/providers/Microsoft.Network/virtualNetworks/vnet1-network/subnets/internal]  azurerm\_network\_interface.main[0]: Creating...  azurerm\_network\_interface.main[1]: Creating...  azurerm\_subnet\_network\_security\_group\_association.main: Creating...  azurerm\_subnet\_network\_security\_group\_association.main: Creation complete after 3s [id=/subscriptions/6ca1dc7b-a8e2-47be-b9ef-0603f8007f95/resourceGroups/Project-Rg1/providers/Microsoft.Network/virtualNetworks/vnet1-network/subnets/internal]  azurerm\_network\_interface.main[1]: Creation complete after 4s [id=/subscriptions/6ca1dc7b-a8e2-47be-b9ef-0603f8007f95/resourceGroups/Project-Rg1/providers/Microsoft.Network/networkInterfaces/Jenkins-vm-1-nic]  azurerm\_network\_interface.main[0]: Creation complete after 4s [id=/subscriptions/6ca1dc7b-a8e2-47be-b9ef-0603f8007f95/resourceGroups/Project-Rg1/providers/Microsoft.Network/networkInterfaces/Jenkins-vm-0-nic]  azurerm\_linux\_virtual\_machine.main[0]: Creating...  azurerm\_linux\_virtual\_machine.main[1]: Creating...  azurerm\_linux\_virtual\_machine.main[0]: Still creating... [10s elapsed]  azurerm\_linux\_virtual\_machine.main[1]: Still creating... [10s elapsed]  azurerm\_linux\_virtual\_machine.main[0]: Still creating... [20s elapsed]  azurerm\_linux\_virtual\_machine.main[1]: Still creating... [20s elapsed]  azurerm\_linux\_virtual\_machine.main[0]: Still creating... [30s elapsed]  azurerm\_linux\_virtual\_machine.main[1]: Still creating... [30s elapsed]  azurerm\_linux\_virtual\_machine.main[0]: Still creating... [40s elapsed]  azurerm\_linux\_virtual\_machine.main[1]: Still creating... [40s elapsed]  azurerm\_linux\_virtual\_machine.main[0]: Creation complete after 47s [id=/subscriptions/6ca1dc7b-a8e2-47be-b9ef-0603f8007f95/resourceGroups/Project-Rg1/providers/Microsoft.Compute/virtualMachines/Jenkins-vm-0]  azurerm\_linux\_virtual\_machine.main[1]: Creation complete after 47s [id=/subscriptions/6ca1dc7b-a8e2-47be-b9ef-0603f8007f95/resourceGroups/Project-Rg1/providers/Microsoft.Compute/virtualMachines/Jenkins-vm-1]  null\_resource.nr1: Creating...  null\_resource.nr2: Creating...  null\_resource.nr1: Provisioning with 'remote-exec'...  null\_resource.nr1 (remote-exec): Connecting to remote host via SSH...  null\_resource.nr1 (remote-exec): Host: 20.120.115.188  null\_resource.nr1 (remote-exec): User: vmadmin  null\_resource.nr1 (remote-exec): Password: true  null\_resource.nr1 (remote-exec): Private key: false  null\_resource.nr1 (remote-exec): Certificate: false  null\_resource.nr1 (remote-exec): SSH Agent: false  null\_resource.nr1 (remote-exec): Checking Host Key: false  null\_resource.nr1 (remote-exec): Target Platform: unix  null\_resource.nr2: Provisioning with 'remote-exec'...  null\_resource.nr2 (remote-exec): Connecting to remote host via SSH...  null\_resource.nr2 (remote-exec): Host: 20.120.115.189  null\_resource.nr2 (remote-exec): User: vmadmin  null\_resource.nr2 (remote-exec): Password: true  null\_resource.nr2 (remote-exec): Private key: false  null\_resource.nr2 (remote-exec): Certificate: false  null\_resource.nr2 (remote-exec): SSH Agent: false  null\_resource.nr2 (remote-exec): Checking Host Key: false  null\_resource.nr2 (remote-exec): Target Platform: unix  null\_resource.nr1 (remote-exec): Connecting to remote host via SSH...  null\_resource.nr1 (remote-exec): Host: 20.120.115.188  null\_resource.nr1 (remote-exec): User: vmadmin  null\_resource.nr1 (remote-exec): Password: true  null\_resource.nr1 (remote-exec): Private key: false  null\_resource.nr1 (remote-exec): Certificate: false  null\_resource.nr1 (remote-exec): SSH Agent: false  null\_resource.nr1 (remote-exec): Checking Host Key: false  null\_resource.nr1 (remote-exec): Target Platform: unix  null\_resource.nr2 (remote-exec): Connecting to remote host via SSH...  null\_resource.nr2 (remote-exec): Host: 20.120.115.189  null\_resource.nr2 (remote-exec): User: vmadmin  null\_resource.nr2 (remote-exec): Password: true  null\_resource.nr2 (remote-exec): Private key: false  null\_resource.nr2 (remote-exec): Certificate: false  null\_resource.nr2 (remote-exec): SSH Agent: false  null\_resource.nr2 (remote-exec): Checking Host Key: false  null\_resource.nr2 (remote-exec): Target Platform: unix  null\_resource.nr1 (remote-exec): Connecting to remote host via SSH...  null\_resource.nr1 (remote-exec): Host: 20.120.115.188  null\_resource.nr1 (remote-exec): User: vmadmin  null\_resource.nr1 (remote-exec): Password: true  null\_resource.nr1 (remote-exec): Private key: false  null\_resource.nr1 (remote-exec): Certificate: false  null\_resource.nr1 (remote-exec): SSH Agent: false  null\_resource.nr1 (remote-exec): Checking Host Key: false  null\_resource.nr1 (remote-exec): Target Platform: unix  null\_resource.nr2 (remote-exec): Connecting to remote host via SSH...  null\_resource.nr2 (remote-exec): Host: 20.120.115.189  null\_resource.nr2 (remote-exec): User: vmadmin  null\_resource.nr2 (remote-exec): Password: true  null\_resource.nr2 (remote-exec): Private key: false  null\_resource.nr2 (remote-exec): Certificate: false  null\_resource.nr2 (remote-exec): SSH Agent: false  null\_resource.nr2 (remote-exec): Checking Host Key: false  null\_resource.nr2 (remote-exec): Target Platform: unix  null\_resource.nr2: Still creating... [10s elapsed]  null\_resource.nr1: Still creating... [10s elapsed]  null\_resource.nr2 (remote-exec): Connecting to remote host via SSH...  null\_resource.nr2 (remote-exec): Host: 20.120.115.189  null\_resource.nr2 (remote-exec): User: vmadmin  null\_resource.nr2 (remote-exec): Password: true  null\_resource.nr2 (remote-exec): Private key: false  null\_resource.nr2 (remote-exec): Certificate: false  null\_resource.nr2 (remote-exec): SSH Agent: false  null\_resource.nr2 (remote-exec): Checking Host Key: false  null\_resource.nr2 (remote-exec): Target Platform: unix  null\_resource.nr1 (remote-exec): Connecting to remote host via SSH...  null\_resource.nr1 (remote-exec): Host: 20.120.115.188  null\_resource.nr1 (remote-exec): User: vmadmin  null\_resource.nr1 (remote-exec): Password: true  null\_resource.nr1 (remote-exec): Private key: false  null\_resource.nr1 (remote-exec): Certificate: false  null\_resource.nr1 (remote-exec): SSH Agent: false  null\_resource.nr1 (remote-exec): Checking Host Key: false  null\_resource.nr1 (remote-exec): Target Platform: unix |

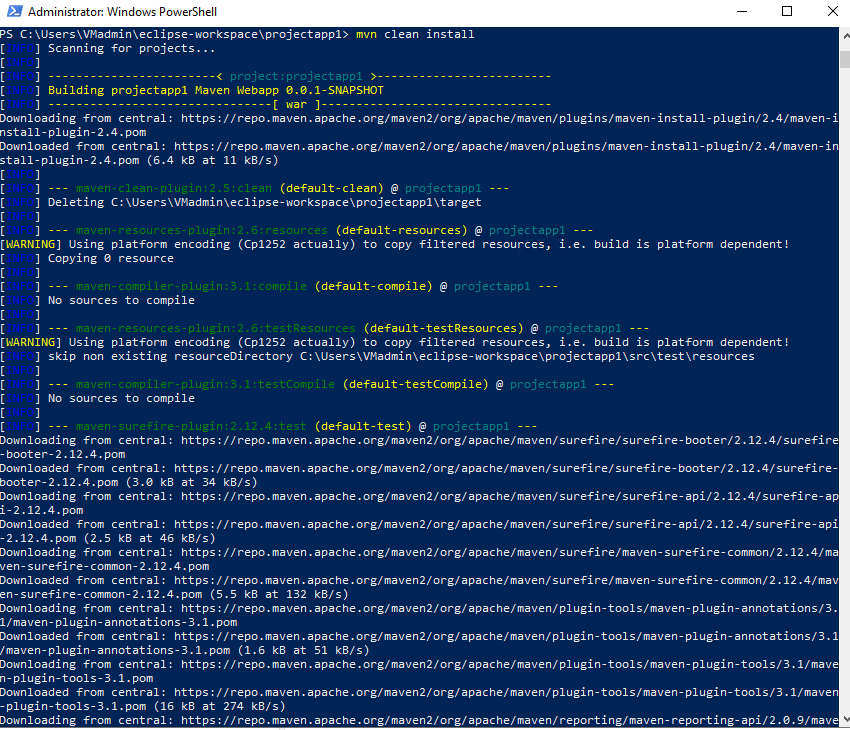
|  |
| --- |
| null\_resource.nr2 (remote-exec): Reading state information... 0%  null\_resource.nr2 (remote-exec): Reading state information... Done  null\_resource.nr2 (remote-exec): The following package was automatically installed and is no longer required:  null\_resource.nr2 (remote-exec): linux-headers-4.15.0-176  null\_resource.nr2 (remote-exec): Use 'sudo apt autoremove' to remove it.  null\_resource.nr2 (remote-exec): Suggested packages:  null\_resource.nr2 (remote-exec): git-daemon-run | git-daemon-sysvinit  null\_resource.nr2 (remote-exec): git-doc git-el git-email git-gui  null\_resource.nr2 (remote-exec): gitk gitweb git-cvs git-mediawiki  null\_resource.nr2 (remote-exec): git-svn  null\_resource.nr2 (remote-exec): The following packages will be upgraded:  null\_resource.nr2 (remote-exec): git  null\_resource.nr2 (remote-exec): 1 upgraded, 0 newly installed, 0 to remove and 10 not upgraded.  null\_resource.nr2 (remote-exec): Need to get 3925 kB of archives.  null\_resource.nr2 (remote-exec): After this operation, 4096 B of additional disk space will be used.  null\_resource.nr2 (remote-exec):  null\_resource.nr2 (remote-exec): 0% [Working]  null\_resource.nr2 (remote-exec): Get:1 http://azure.archive.ubuntu.com/ubuntu bionic-updates/main amd64 git amd64 1:2.17.1-1ubuntu0.11 [3925 kB]  null\_resource.nr2 (remote-exec):  null\_resource.nr2 (remote-exec): 0% [1 git 0 B/3925 kB 0%]  null\_resource.nr2 (remote-exec): 100% [Working]  null\_resource.nr2 (remote-exec): Fetched 3925 kB in 0s (35.8 MB/s)  (Reading database ...  null\_resource.nr2 (remote-exec): (Reading database ... 5%  null\_resource.nr2 (remote-exec): (Reading database ... 10%  null\_resource.nr2 (remote-exec): (Reading database ... 15%  null\_resource.nr2 (remote-exec): (Reading database ... 20%  null\_resource.nr2 (remote-exec): (Reading database ... 25%  null\_resource.nr2 (remote-exec): (Reading database ... 30%  null\_resource.nr2 (remote-exec): (Reading database ... 35%  null\_resource.nr2 (remote-exec): (Reading database ... 40%  null\_resource.nr2 (remote-exec): (Reading database ... 45%  null\_resource.nr2 (remote-exec): (Reading database ... 50%  null\_resource.nr2 (remote-exec): (Reading database ... 55%  null\_resource.nr2 (remote-exec): (Reading database ... 60%  null\_resource.nr2 (remote-exec): (Reading database ... 65%  null\_resource.nr2 (remote-exec): (Reading database ... 70%  null\_resource.nr2 (remote-exec): (Reading database ... 75%  null\_resource.nr2 (remote-exec): (Reading database ... 80%  null\_resource.nr2 (remote-exec): (Reading database ... 85%  null\_resource.nr2 (remote-exec): (Reading database ... 90%  null\_resource.nr2 (remote-exec): (Reading database ... 95%  null\_resource.nr2 (remote-exec): (Reading database ... 100%  null\_resource.nr2 (remote-exec): (Reading database ... 139138 files and directories currently installed.)  null\_resource.nr2 (remote-exec): Preparing to unpack .../git\_1%3a2.17.1-1ubuntu0.11\_amd64.deb ...  Progress: [ 17%] [##.............Unpacking git (1:2.17.1-1ubuntu0.11) over (1:2.17.1-1ubuntu0.10) ...  Progress: [ 50%] [#########.........]  null\_resource.nr2 (remote-exec): Setting up git (1:2.17.1-1ubuntu0.11) ...  Progress: [ 83%] [###############...]  null\_resource.nr2: Creation complete after 5m46s [id=793273962]  Apply complete! Resources: 13 added, 0 changed, 0 destroyed.  Outputs:  public\_ip\_address = [  "",  "",  ]  PS C:\tfdemos\project> |

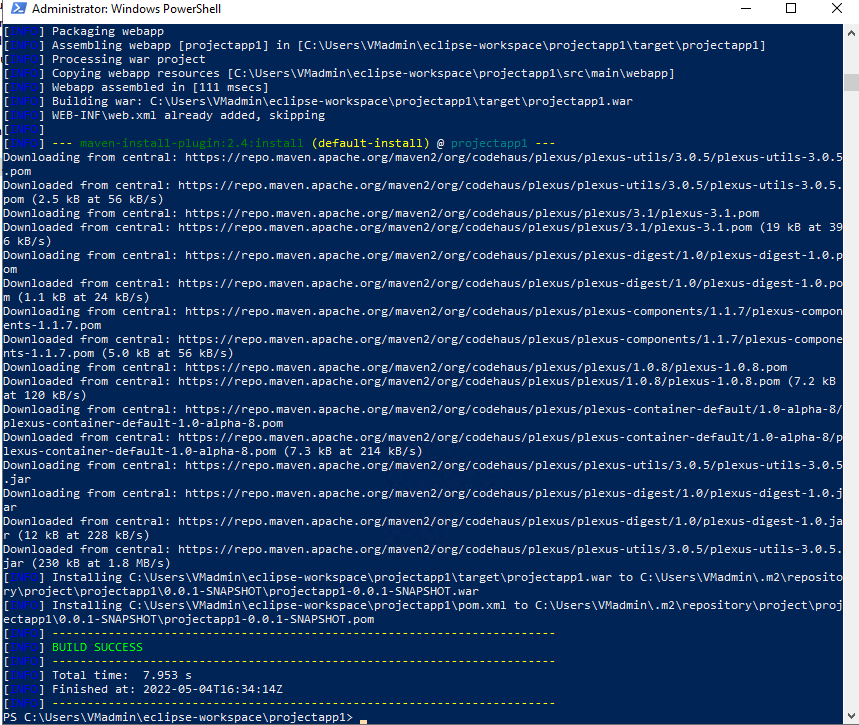


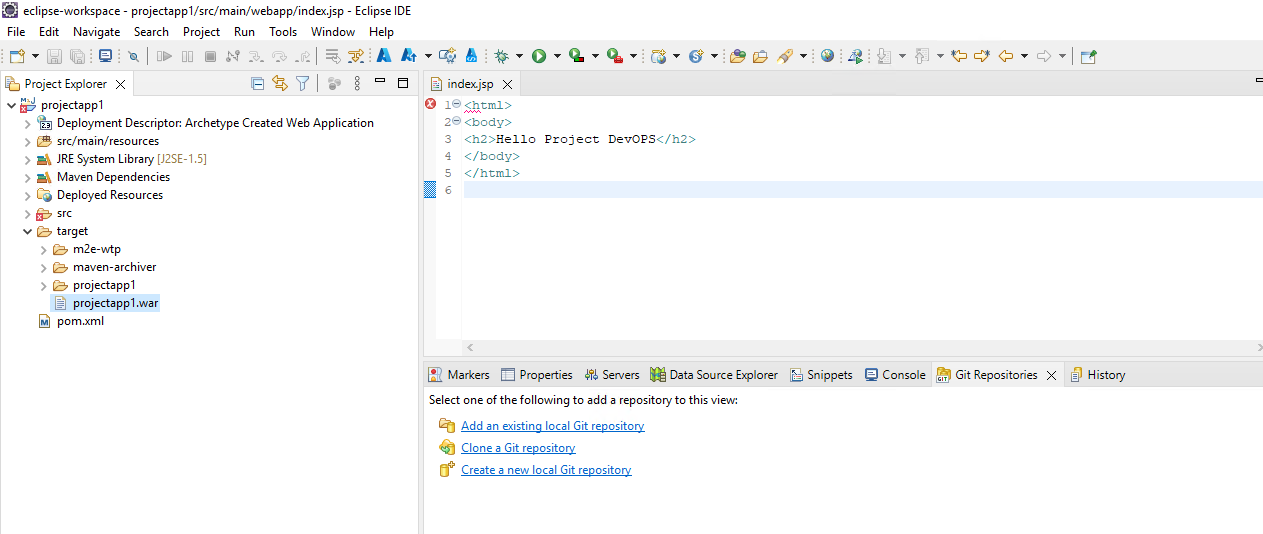


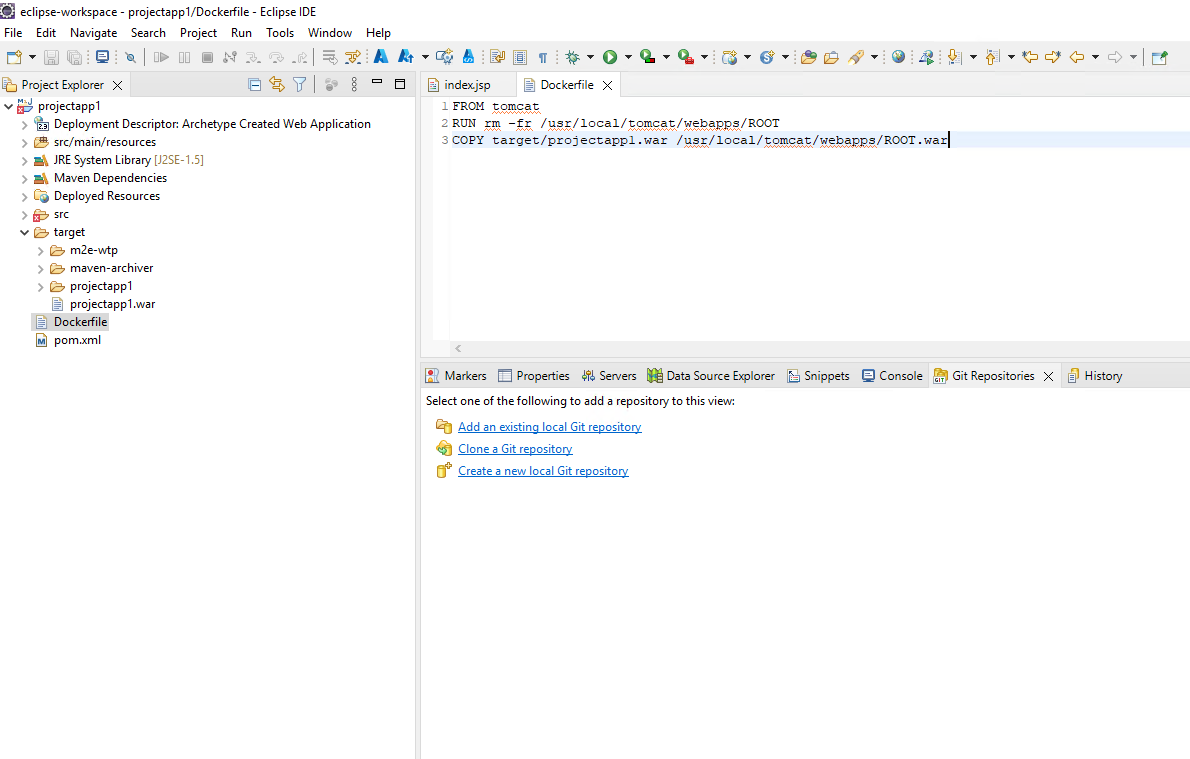








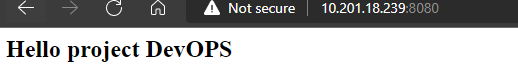


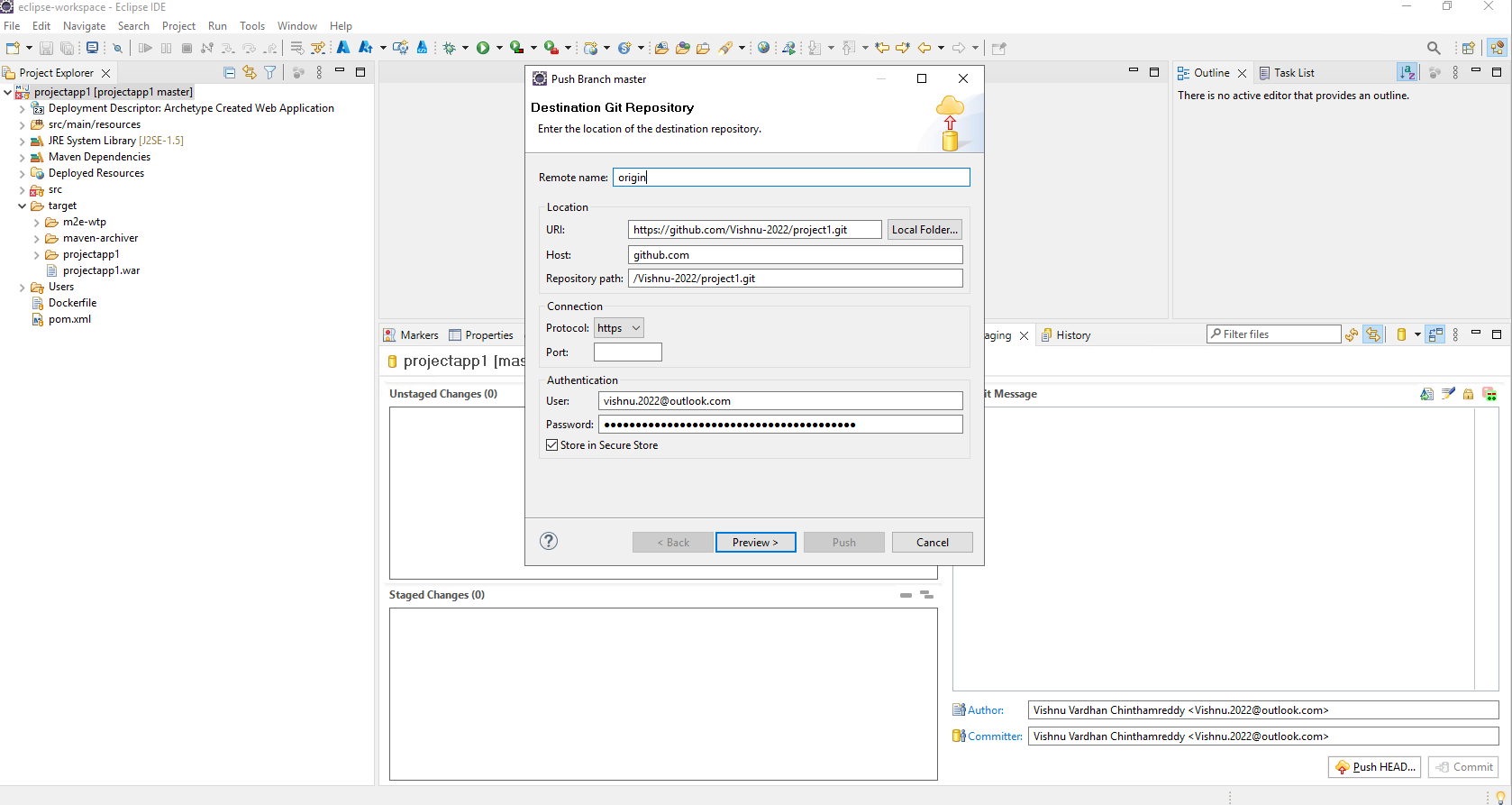


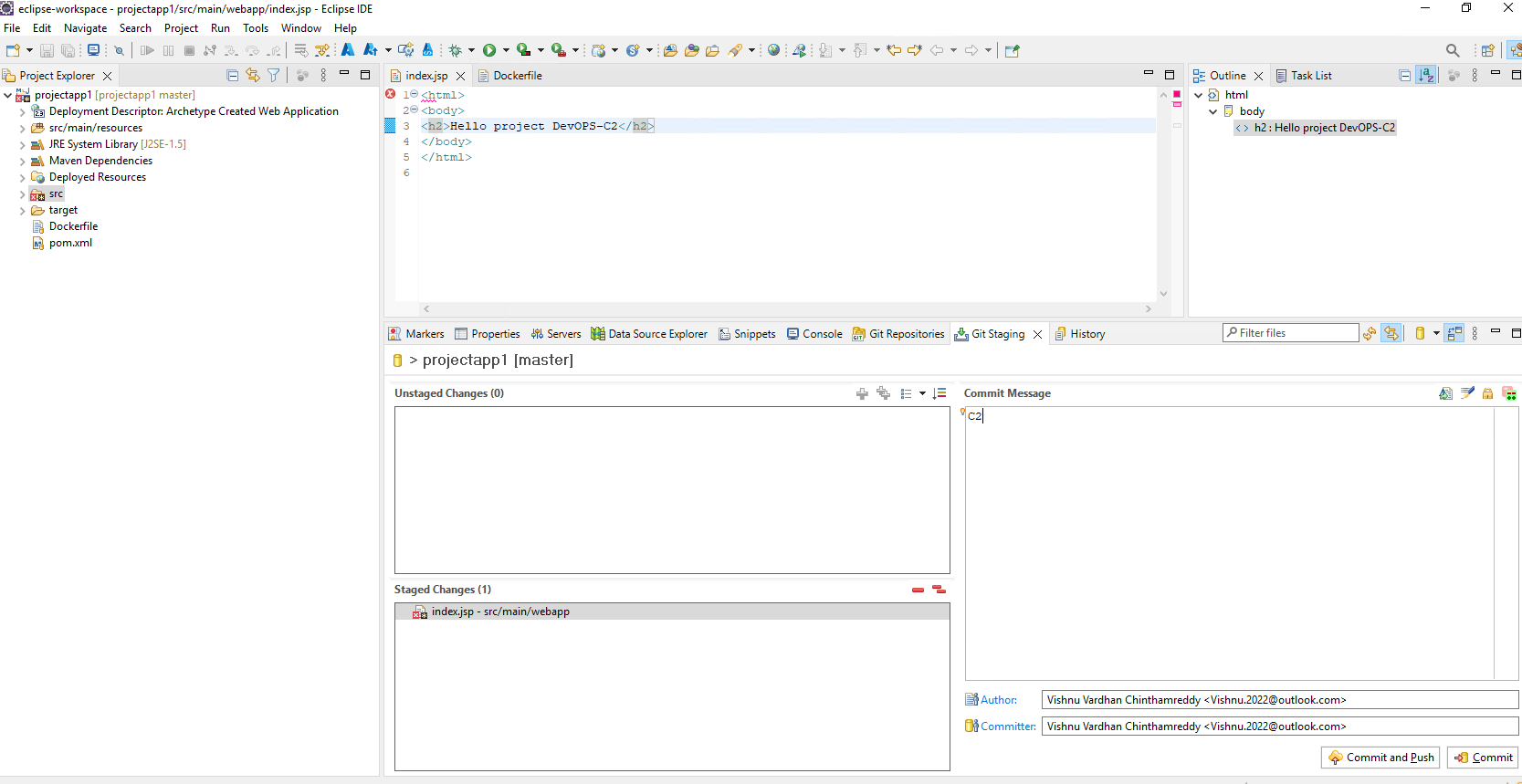
**Tested Locally the Image**

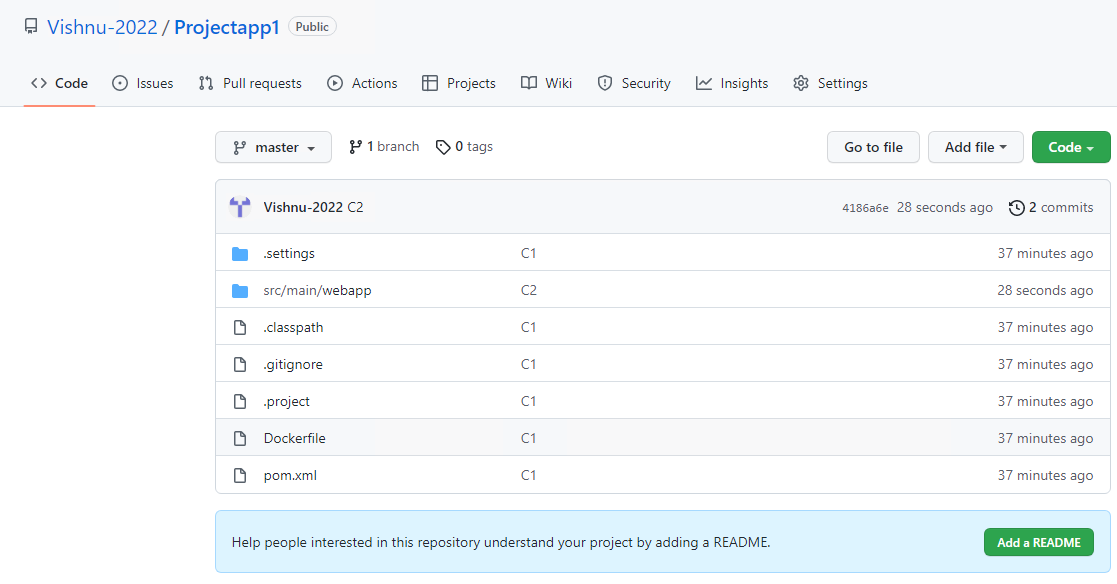


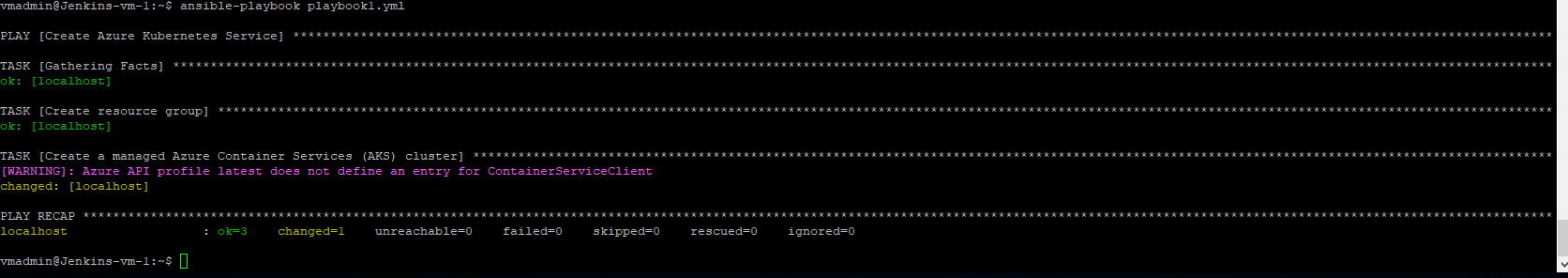


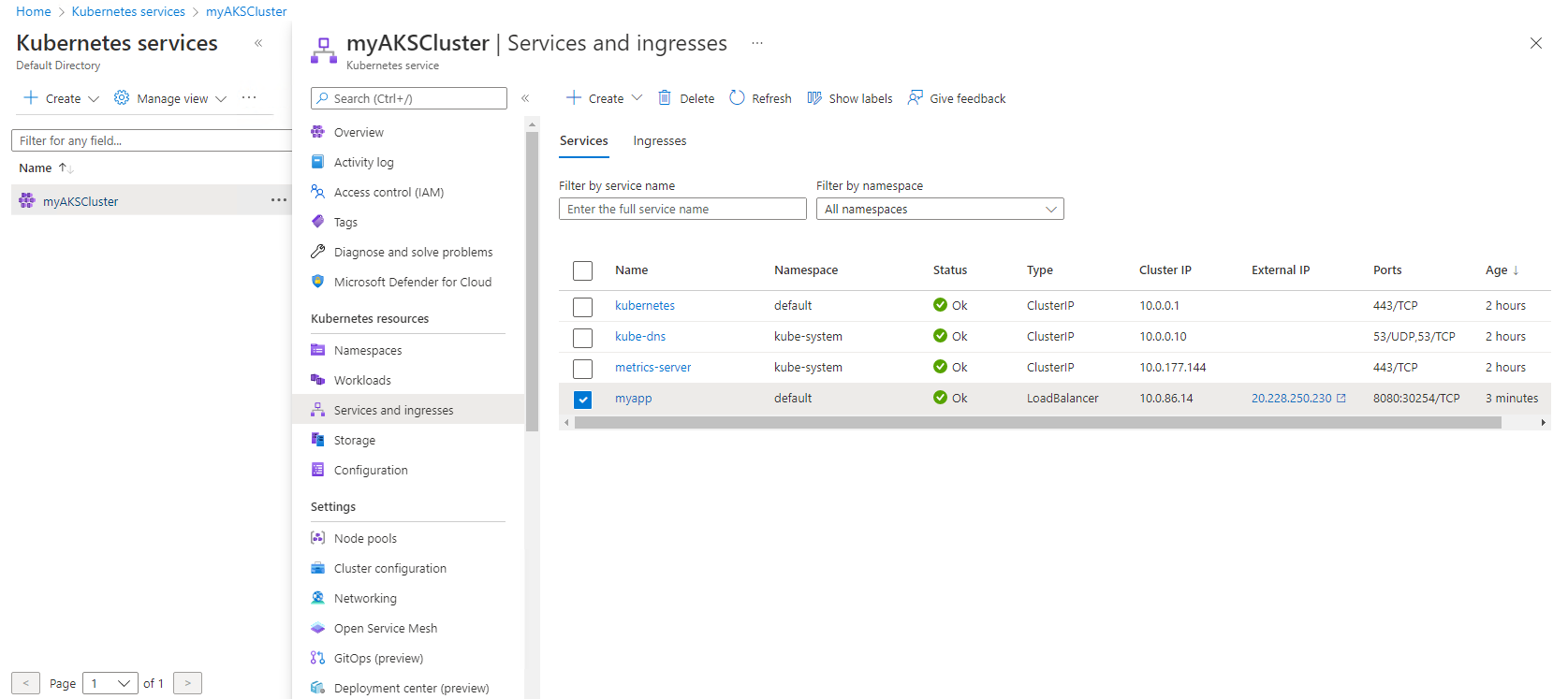


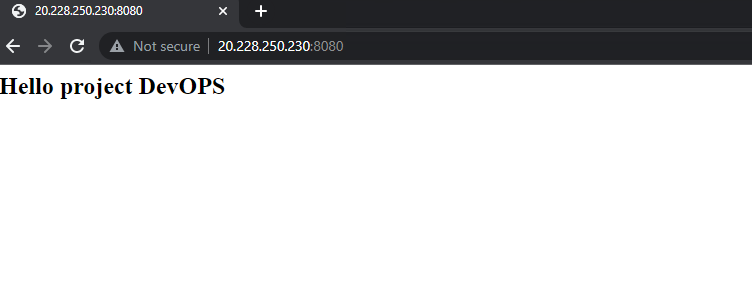


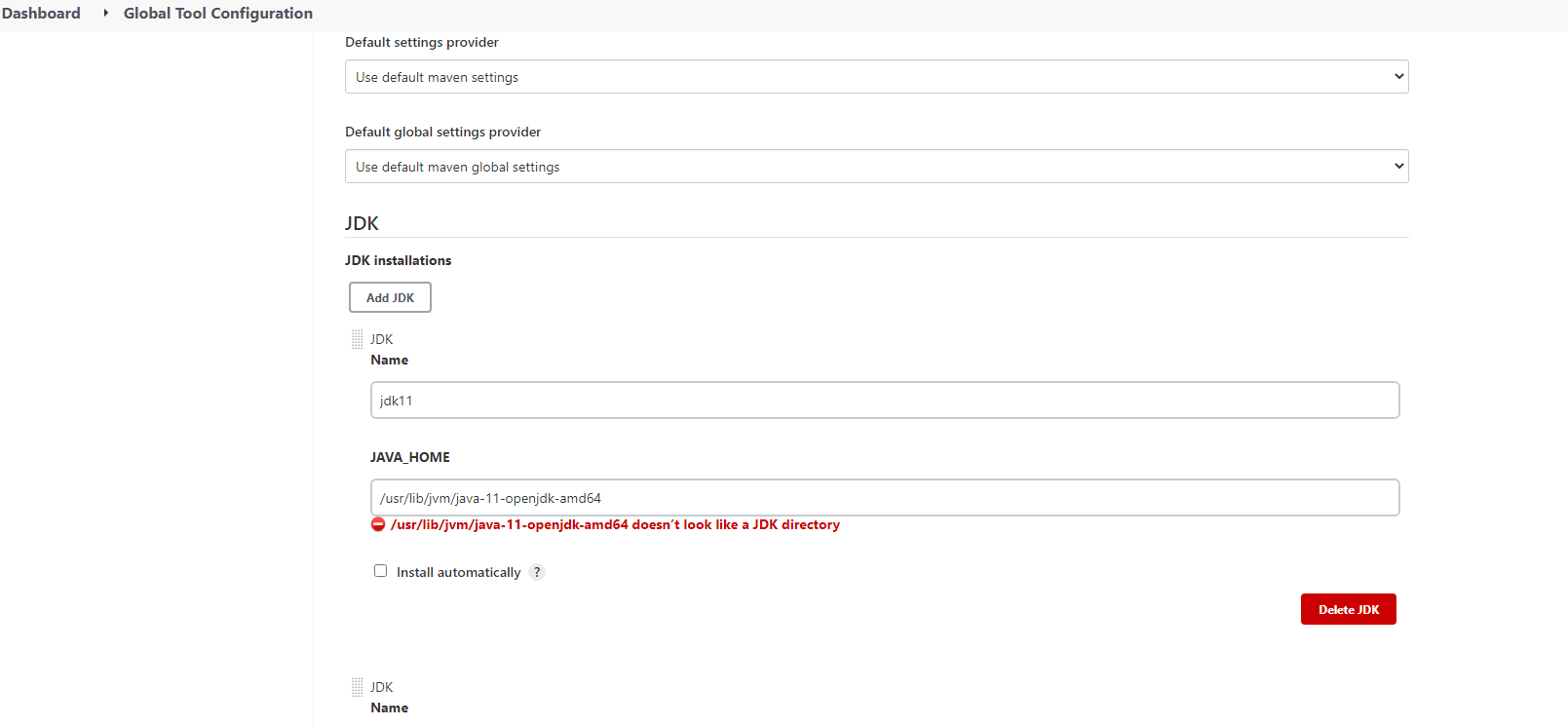


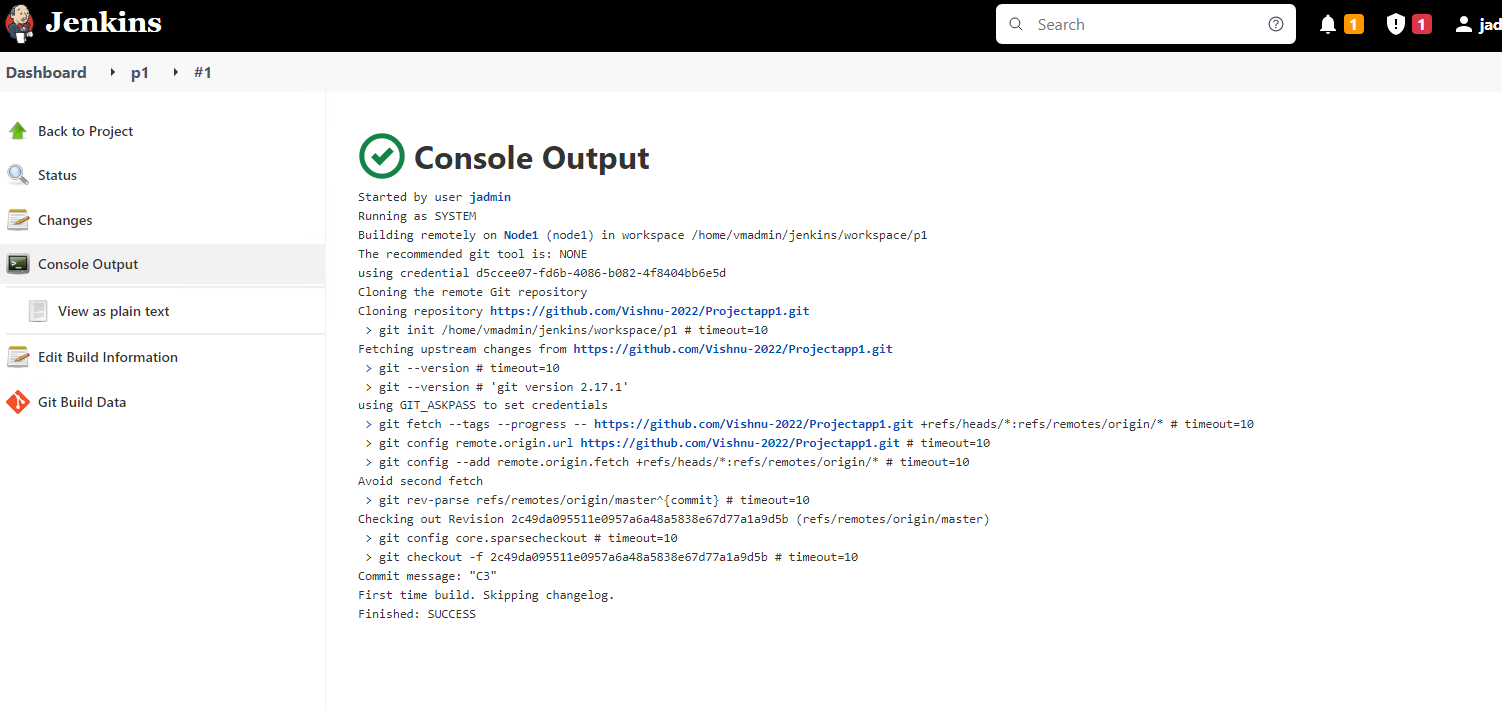




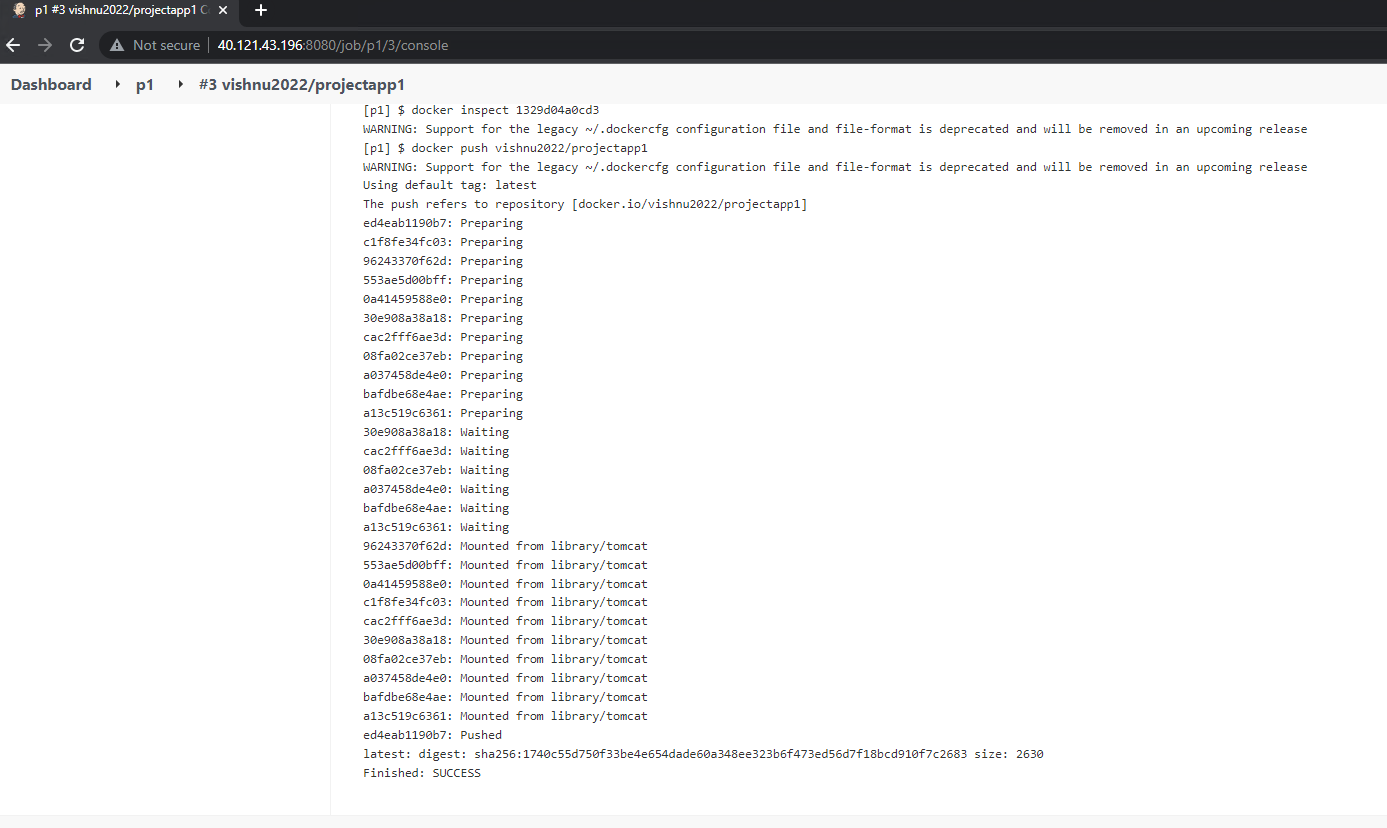


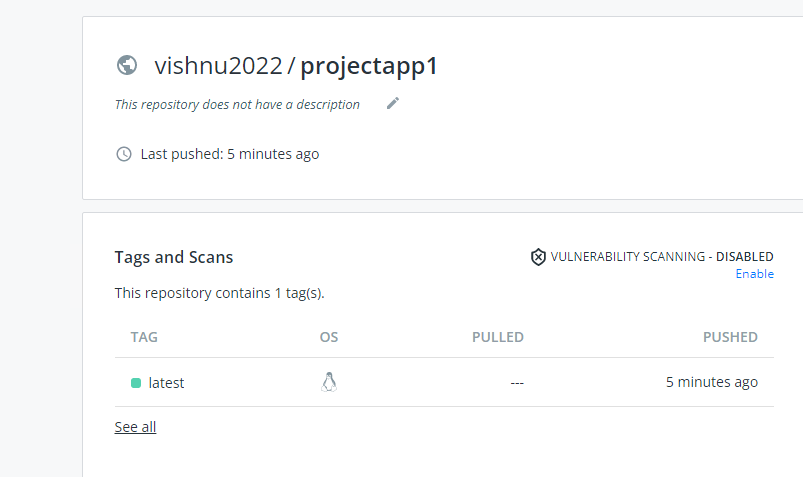


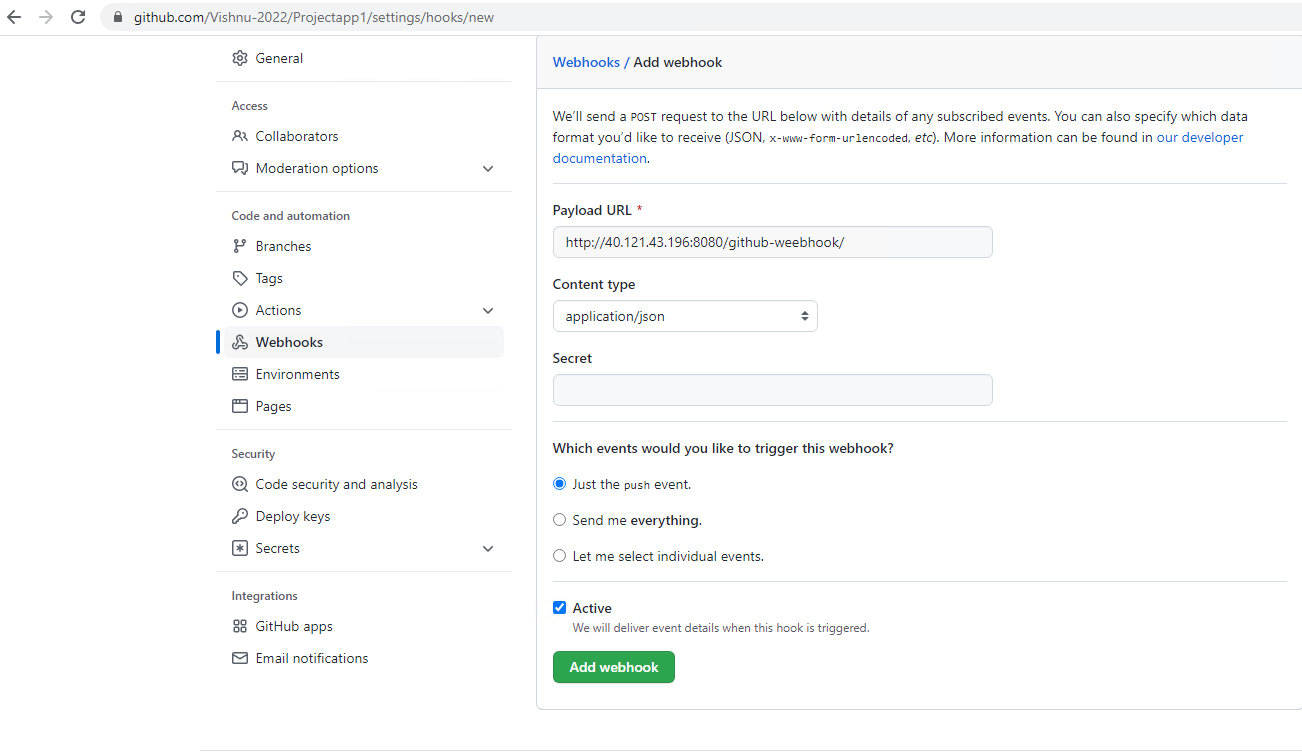


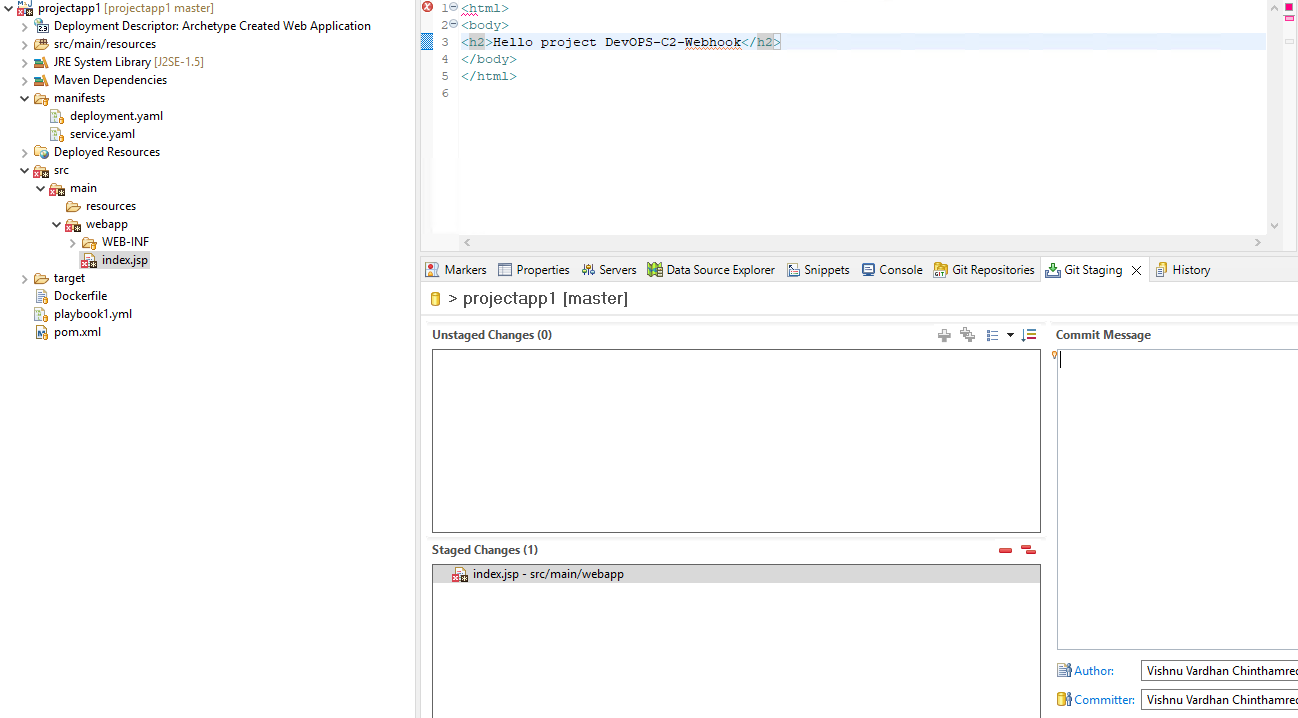


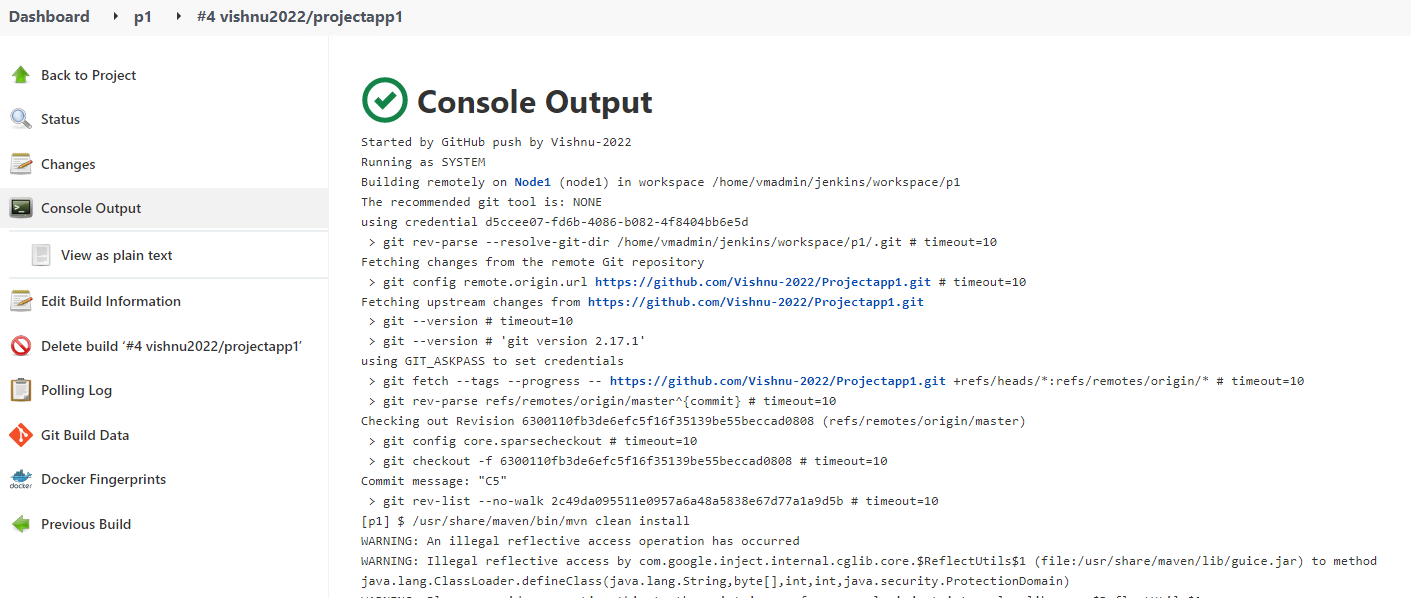


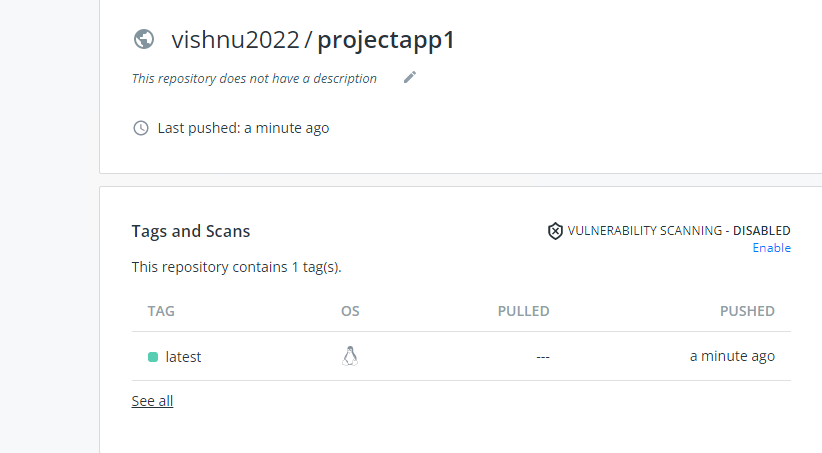


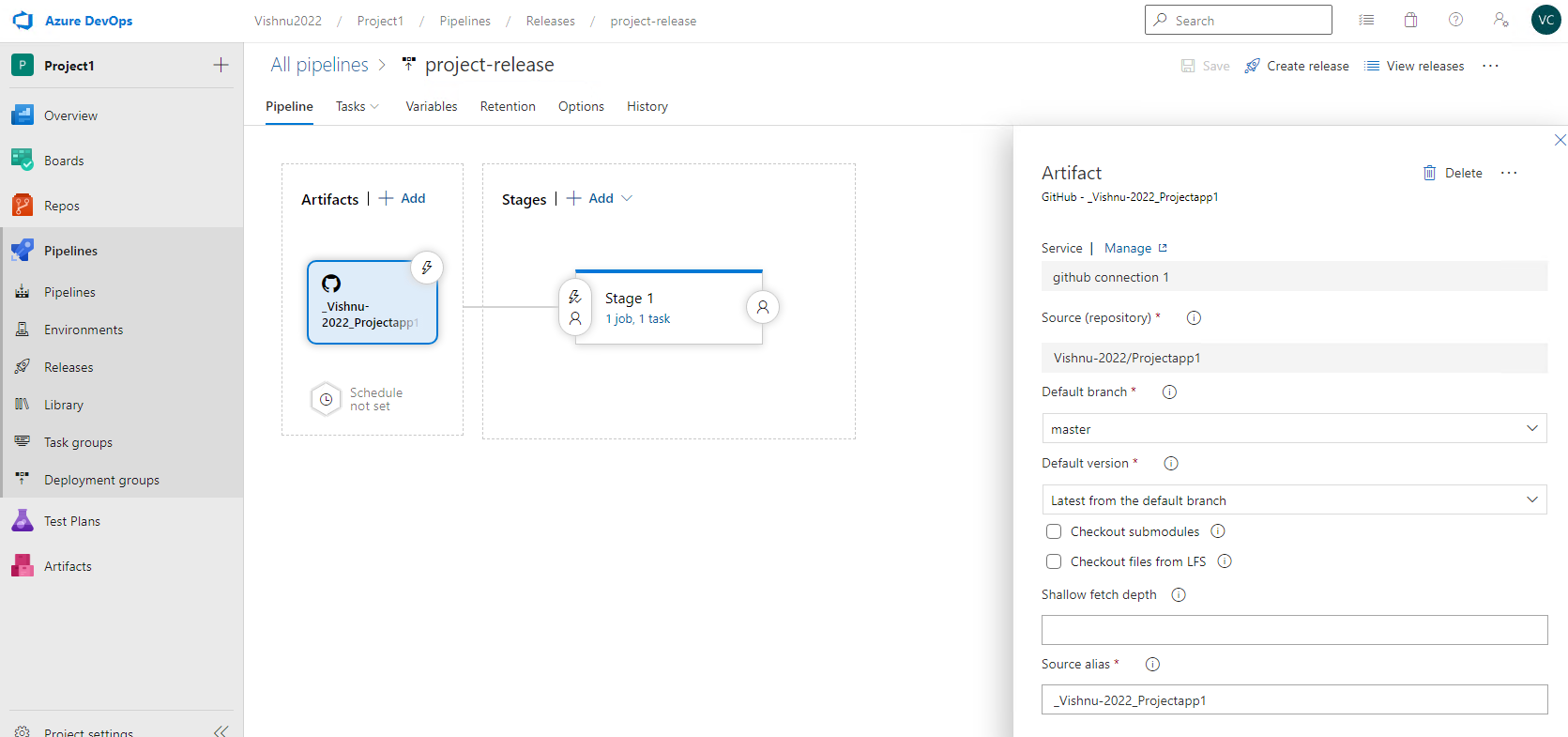


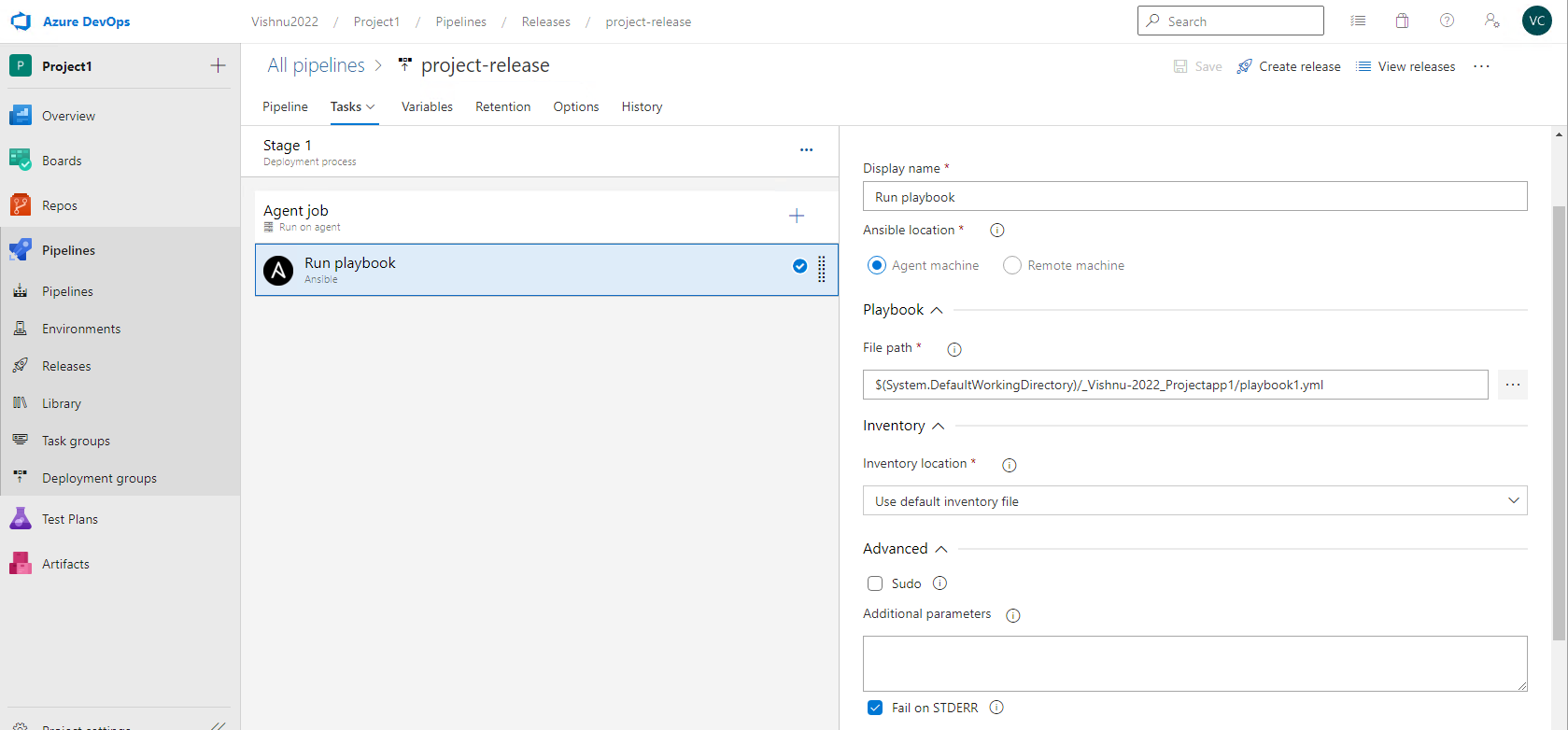












**Summary:**

Completed all the steps except

In Build Stage Step2 : Deploy the pods to AKS Cluster using image created in Docker HUB or ACR – However it worked when deployed Manually from Agent using Azure repo repo9:projectapp1

Thank you

Regards,

Vishnu Vardhan