

ASSIGNMENT 41

Automate the Book Review Application with Terraform, Ansible, and Azure DevOps

This assignment focuses on automating the deployment of the Book Review Application using Terraform, Ansible, and Azure DevOps. It combines infrastructure provisioning and application configuration into a seamless CI/CD workflow. This approach is highly useful because it ensures consistent, repeatable deployments, reduces manual errors, and speeds up delivery. By integrating Terraform and Ansible within Azure DevOps pipelines, teams can achieve full end-to-end automation — from setting up servers to deploying and updating the application with minimal effort.

Objective

Reproduce an end-to-end automation, using two repositories and two Azure DevOps pipelines to deploy the Book Review App:

- Infra repository (Terraform): provisions the infrastructure (resource group/VNet/2 VMs/MySQL).
- App repository (Book Review App + Ansible): configures frontend and backend VMs and deploys the application.

Application repo to deploy: <https://github.com/pravinmishraaws/book-review-app>

Note everything was done in my agent machine

Step 1: The Build

Repositories

1. Book-review-infra <https://github.com/pravinmishraaws/book-review-infra>
 - Terraform code to provision:
 - 1 frontend VM (Ubuntu)
 - 1 backend VM (Ubuntu)
 - VNet / subnet
 - MySQL (Azure MySQL or VM-based; follow the video)
 - Terraform outputs:
 - frontend_public_ip
 - backend_public_ip
 - mysql_fqdn (or endpoint)
2. Book-review-app <https://github.com/pravinmishraaws/book-review-app>
 - App source code (frontend + backend)
 - Ansible playbooks to:
 - configure common packages
 - configure backend (API, DB connection)
 - configure frontend (points to backend)
 - restart services / Nginx

Pipelines

1. Infra pipeline (Azure Pipelines):
 - Source: *book-review-infra* (in GitHub or Azure Repos)
 - Install Terraform
 - Authenticate using Azure Resource Manager service connection (SPN)
 - *terraform init/plan/apply*
 - Print/log Terraform outputs (frontend IP, backend IP, DB FQDN)
 - This is the “*platform / infra*” pipeline
2. App pipeline (Azure Pipelines):
 - Source: *book-review-app* repo
 - Install Ansible
 - Download SSH private key from Secure Files
 - Update Ansible inventory and vars manually with values from the infra pipeline
 - Run Ansible playbook to configure both VMs and deploy the app
 - This is the “*application / Dev*” pipeline

Step 2: Background Setup:

1. Install terraform

```
sudo apt update && sudo apt install -y wget curl unzip
```

```
sudo apt-get update && sudo apt-get install -y gnupg software-properties-common
```

```
wget -O- https://apt.releases.hashicorp.com/gpg | gpg --dearmor | sudo tee /usr/share/keyrings/hashicorp-archive-keyring.gpg
```

```
echo "deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] https://apt.releases.hashicorp.com $(lsb_release -cs) main" | sudo tee /etc/apt/sources.list.d/hashicorp.list
```

```
sudo apt update
```

```
sudo apt install terraform
```

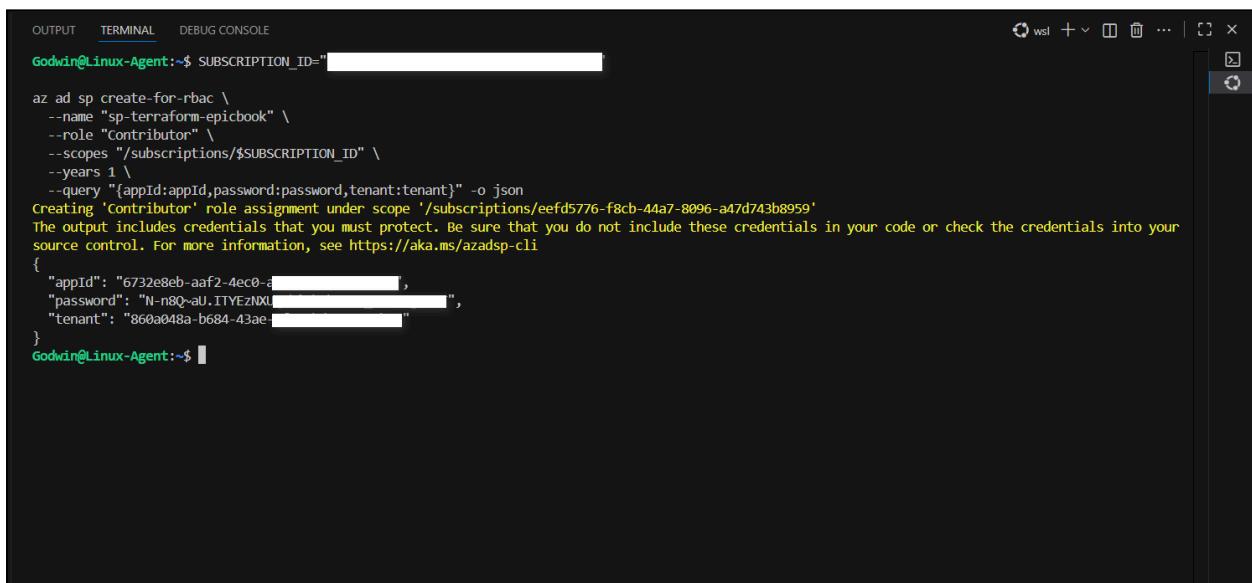
```
terraform -version
```

2. Download AZ in linux

```
curl -sL https://aka.ms/InstallAzureCLIDeb | sudo bash
```

3. az login

4 Create Service Principle



```
Godwin@Linux-Agent:~$ SUBSCRIPTION_ID="REDACTED"
az ad sp create-for-rbac \
--name "sp-terraform-epicbook" \
--role "Contributor" \
--scopes "/subscriptions/$SUBSCRIPTION_ID" \
--years 1 \
--query "[appId:appId,password:password,tenant:tenant]" -o json
Creating 'Contributor' role assignment under scope '/subscriptions/eefdf5776-f8cb-44a7-8096-a47d743b8959'
The output includes credentials that you must protect. Be sure that you do not include these credentials in your code or check the credentials into your source control. For more information, see https://aka.ms/azadsp-cli
{
  "appId": "6732e8eb-aaf2-4ec0-a[REDACTED]",
  "password": "N-n8Q-aU.IIVYeNxU[REDACTED]",
  "tenant": "860a048a-b684-43ae-[REDACTED]"
}
Godwin@Linux-Agent:~$
```

5. Azure Resource Manager

Create an Azure Resource Manager service connection using the SPN/App Registration

In azure DevOps, go to book review project settings, create service connections. Don't use SSH. Use Azure Resource

The screenshot shows the 'Project Settings' page for a 'BookReview' project. A red box highlights the 'Service connections' link in the left sidebar. A red box also highlights the 'Create service connection' button in the center of the page. A large red box encloses the 'New Azure service connection' dialog box.

New Azure service connection

Identity type: App registration or managed identity (manual)

Credential: Secret

Environment: Azure Cloud

Scope Level: Subscription

Subscription ID: [redacted]

Subscription name: Sub1

Authentication: Refer to [App registration](#) or [Managed Identity](#) configuration

The screenshot shows the 'Service connections' page for the 'BookReview' project. A red box highlights the search bar with the text 'bookreview-connect'. The page lists one service connection: 'bookreview-connect'.

6 Keygen

Upload your keygen (both private and public) to Azure DevOps-Pipeline-Library.

The screenshot shows the 'Library' page under 'Pipelines' in the 'BookReview' project. A red box highlights the 'Secure files' tab. A red box highlights the 'Secure file' button. A red box highlights the list of uploaded files: 'id_rsa' and 'id_rsa.pub'.

Name	Date modified	Modified by	Description
id_rsa	just now	Godwin Eyo	
id_rsa.pub	just now	Godwin Eyo	

Step 3: App Repository, Infra Repository and Pipeline

1. Clone/import the book-review-app

The screenshot shows two consecutive views of the Azure DevOps interface.

Top Screenshot: The "Import a Git repository" dialog is open. The "Repository type" dropdown is set to "Git". The "Clone URL" field contains "https://github.com/pravinmishraaws/book-review-app". The "Name" field is set to "book-review-app". The "Import" button is visible at the bottom right.

Bottom Screenshot: The repository structure is displayed under "Files". The root directory "book-review-app" contains subfolders "ansible", "backend", and "frontend", along with files ".DS_Store", "azure-pipeline.yaml", "Docker_README.md", "docker-compose.yml", "mysql_docker.md", and "README.md". A table lists the contents of the "main" branch, showing commit details for each file.

Name	Last change	Commits
ansible	Yesterday	95c352bc Updated backend.yml Godwin Eyo
backend	Apr 29	da3897ed Update README.md Pravin Mishra
frontend	Mar 18	df5f03fc Update docker_frontend-README.md Pravin Mishra
.DS_Store	May 2	f5af4c71 adding cicd pipeline and ansible code Pravin Mishra
azure-pipeline.yaml	Yesterday	ead9438f Updated azure-pipeline.yaml Godwin Eyo
Docker_README.md	May 14	52b90632 Update docker_compose_README.md Pravin Mishra
docker-compose.yml	May 14	c0e44b81 Update docker-compose.yml Pravin Mishra
mysql_docker.md	Mar 18	182cc3da Create mysql_docker.md Pravin Mishra
README.md	May 8	cc897cb1 testing automated CI/CD Pravin Mishra

2. Clone/import *book-review-infra* and place the Terraform code (modules for network, compute, database; separate env folders).

3. Create an Azure Pipeline (YAML) in Azure DevOps for this repo:

- Use self-hosted agent (preferred)
- Install Terraform
- Download SSH public key from Secure Files and place it where Terraform expects it
- Run *terraform init*, *terraform plan*, and *terraform apply*

The screenshot shows the Azure DevOps interface for the 'BookReview' project. On the left, the 'Repos' section is selected, and the 'Files' tab is highlighted. The main area displays the contents of the 'book-review-app' repository, including files like ansible, backend, frontend, and various configuration and Docker files. A modal window titled 'Import a Git repository' is open, prompting for a 'Clone URL' (set to https://github.com/pravinmishraaws/book-review-infra) and a 'Name' (set to book-review-infra). The 'Import' button at the bottom right of the modal is visible.

The screenshot shows the 'book-review-infra' repository details in the Azure DevOps interface. The repository contains a 'terraform' folder, a '.gitignore' file, an 'azure-pipelines.yaml' file, and a 'README.md' file. The 'README.md' file content includes a note about testing after installing all hosted agent soft and tool. A green 'succeeded' status badge and a 'Clone' button are visible on the right side of the repository card.

The screenshot shows the 'Configure your pipeline' step in the Azure DevOps interface. The 'Pipelines' tab is selected in the sidebar. Under 'New pipeline', the 'Existing Azure Pipelines YAML file' option is highlighted. A modal window titled 'Select an existing YAML file' allows selecting an Azure Pipelines YAML file from the 'book-review-infra' repository. The path '/azure-pipelines.yaml' is selected in the 'Path' dropdown. The 'Continue' button at the bottom right of the modal is visible.

The screenshot shows the Azure DevOps Pipelines interface. On the left, there's a sidebar with 'BookReview' selected. The main area has tabs: 'Connect', 'Select', 'Configure', and 'Review', with 'Review' being the active tab. Below the tabs, it says 'New pipeline' and 'Review your pipeline YAML'. There's a file path 'book-review-infra / azure-pipelines.yaml' with a red box around it. To the right are 'Variables' and 'Run' buttons. The code editor shows the YAML configuration with a red box highlighting the 'pool' section.

```

trigger:
paths:
  include:
    - terraform/**

branches:
  include:
    - main

stages:
  - stage: Terraform
    displayName: 'Provision Infrastructure'
    jobs:
      - job: Terraform_Apply
        displayName: 'Terraform Apply'
        pool:
          name: 'linux-self-agent'
        steps:
          # Not needed for hosted vm
          # - task: UsePythonVersion@0
          #   inputs:
          #     versionSpec: '3.x'

        - task: DownloadSecureFile@1
          name: download_ssh_key
          inputs:
            secureFile: 'id_rsa.pub'

        - script: |
            echo "Copying SSH public key into Terraform module"
            mkdir -p ~/.ssh
            cp $(download_ssh_key.secureFilePath) ~/.ssh/id_rsa.pub
            # Now copy into Terraform module path
            cp $(download_ssh_key.secureFilePath) terraform/modules/compute/id_rsa.pub

        echo "Listing contents of terraform/modules/compute/"
        ls -al terraform/modules/compute/
        displayName: 'Extract SSH Public Key for Terraform'

  - task: AzureCLI@2

```

```

trigger:
paths:
  include:
    - terraform/**

branches:
  include:
    - main

stages:
  - stage: Terraform
    displayName: 'Provision Infrastructure'
    jobs:
      - job: Terraform_Apply
        displayName: 'Terraform Apply'
        pool:
          name: 'linux-self-agent'
        steps:
          # Not needed for hosted vm
          # - task: UsePythonVersion@0
          #   inputs:
          #     versionSpec: '3.x'

        - task: DownloadSecureFile@1
          name: download_ssh_key
          inputs:
            secureFile: 'id_rsa.pub'

        - script: |
            echo "Copying SSH public key into Terraform module"
            mkdir -p ~/.ssh
            cp $(download_ssh_key.secureFilePath) ~/.ssh/id_rsa.pub
            # Now copy into Terraform module path
            cp $(download_ssh_key.secureFilePath) terraform/modules/compute/id_rsa.pub

        echo "Listing contents of terraform/modules/compute/"
        ls -al terraform/modules/compute/
        displayName: 'Extract SSH Public Key for Terraform'

  - task: AzureCLI@2

```

```

inputs:
  azureSubscription: 'bookreview-connect'
  scriptType: bash
  scriptLocation: inlineScript
  workingDirectory: $(System.DefaultWorkingDirectory)/terraform
  inlineScript: |
    echo "##[group]Terraform Init/Plan/Apply"
    cd terraform
    terraform init -input=false
    terraform plan -input=false -var-file="envs/dev.tfvars"
    terraform apply -input=false -auto-approve -var-file="envs/dev.tfvars"
    echo "##[endgroup]"

```

#20251105.1 • Updated azure-pipelines.yaml

Manually run by Godwin Eyo

Repository and version: book-review-infra main 127913b6

Time started and elapsed: Just now

This pipeline needs permission to access 2 resources before this run can continue to Provision Infrastructure.

Jobs

Name	Status
Terraform Apply	Waiting

Jobs in run #20251105.1

Provision Infrastructure

- Terraform Apply (6m 23s)
 - Initialize job (5s)
 - Pre-job: download_ssh... (5s)
 - Checkout book-review... (5s)
 - Extract SSH Public Key... (2s)
 - AzureCLI (6m 1s)
 - Post-job: Checkout boo... (1s)
 - Finalize Job (<1s)
 - Report build status (<1s)

AzureCLI

```

857 module.database.azurevern.mysql.flexible_server.firewall_rule.allow_backend_vm: Still creating... [0m10s elapsed]
858 module.database.azurevern.mysql.flexible_database.bookreviews_db: Still creating... [0m00s elapsed]
859 module.database.azurevern.mysql.flexible_database.bookreviews_db: Creation complete after 18s [id:/subscriptions/.../resourceGroups/.../providers/Microsoft.DBforMySQL/flexibleServers/bookreviews/providers/Microsoft.Network/networkSecurityGroups/bookreview/resourceId=bookreviews_db]
860 module.database.azurevern.mysql.flexible_server.firewall_rule.allow_backend_vm: Still creating... [0m20s elapsed]
861 module.database.azurevern.mysql.flexible_server.firewall_rule.allow_backend_vm: Still creating... [0m40s elapsed]
862 module.database.azurevern.mysql.flexible_server.firewall_rule.allow_backend_vm: Still creating... [0m40s elapsed]
863 module.database.azurevern.mysql.flexible_server.firewall_rule.allow_backend_vm: Still creating... [0m40s elapsed]
864 module.database.azurevern.mysql.flexible_server.firewall_rule.allow_backend_vm: Still creating... [0m40s elapsed]
865 module.database.azurevern.mysql.flexible_server.firewall_rule.allow_backend_vm: Still creating... [0m40s elapsed]
866 module.database.azurevern.mysql.flexible_server.firewall_rule.allow_backend_vm: Still creating... [0m40s elapsed]
867 module.database.azurevern.mysql.flexible_server.firewall_rule.allow_backend_vm: Still creating... [0m40s elapsed]
868 module.database.azurevern.mysql.flexible_server.firewall_rule.allow_backend_vm: Still creating... [0m40s elapsed]
869 module.database.azurevern.mysql.flexible_server.firewall_rule.allow_backend_vm: Still creating... [0m50s elapsed]
870 module.database.azurevern.mysql.flexible_server.firewall_rule.allow_backend_vm: Still creating... [0m50s elapsed]
871 module.database.azurevern.mysql.flexible_server.firewall_rule.allow_backend_vm: Creation complete after 2m45s [id:/subscriptions/.../resourceGroups/.../providers/Microsoft.DBforMySQL/flexibleServers/bookreviews/providers/Microsoft.Network/networkSecurityGroups/bookreview/resourceId=bookreviews_db]
872
873 Apply complete! Resources: 14 added, 0 changed, 0 destroyed.
874
875 Outputs:
876
877 backend_public_ip = "20.248.126.94"
878 frontend_public_ip = "20.28.47.108"
879 mysql_fqdn = "bookreview-db1008.mysql.database.azure.com"
880 nsg_id = "/subscriptions/.../resourceGroups/DevOps1-pm1-dev-rg/providers/Microsoft.Network/networkSecurityGroups/bookreview"
881 public_subnet_id = "/subscriptions/.../resourceGroups/DevOps1-pm1-dev-rg/providers/Microsoft.Network/virtualNetworks/bookre"
882
883 /usr/bin/az account clear
884 Finishing: AzureCLI

```

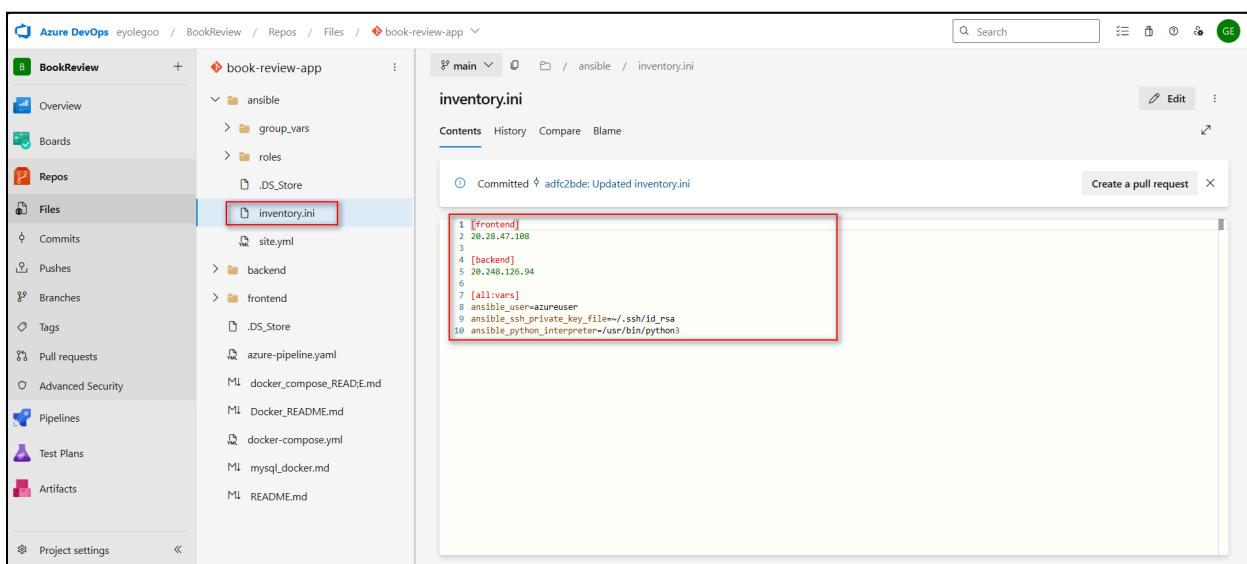
```

backend_public_ip = "20.248.126.94"
frontend_public_ip = "20.28.47.108"
mysql_fqdn = "bookreview-db1008.mysql.database.azure.com"

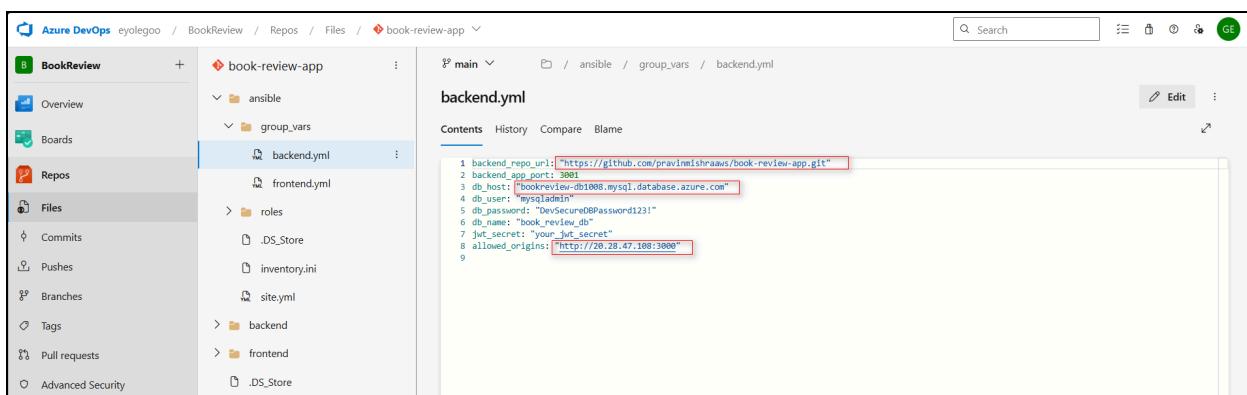
```

Step 4: Manual handoff

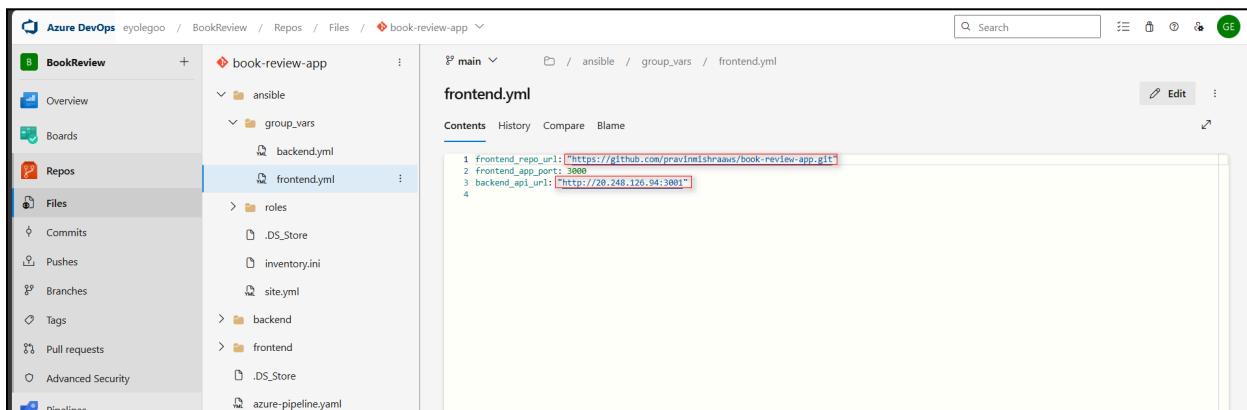
1. From the pipeline logs of the Infra pipeline, copy:
 - Frontend public IP
 - Backend public IP
 - MySQL FQDN / endpoint
2. Open the *book-review-app* repo
3. Update the following files (names may differ depending on your structure):
 - *ansible/inventory.ini* → set frontend and backend hosts to the new IPs
 - *ansible/group_vars/backend.yml* (or similar) → set DB host / FQDN
 - *ansible/group_vars/frontend.yml* (or similar) → set backend API URL



```
1 [frontend]
2 20.28.47.108
3
4 [backend]
5 20.248.126.94
6
7 [allvans]
8 ansible_user=azureuser
9 ansible_ssh_private_key_file=~/.ssh/id_rsa
10 ansible_python_interpreter=/usr/bin/python3
```



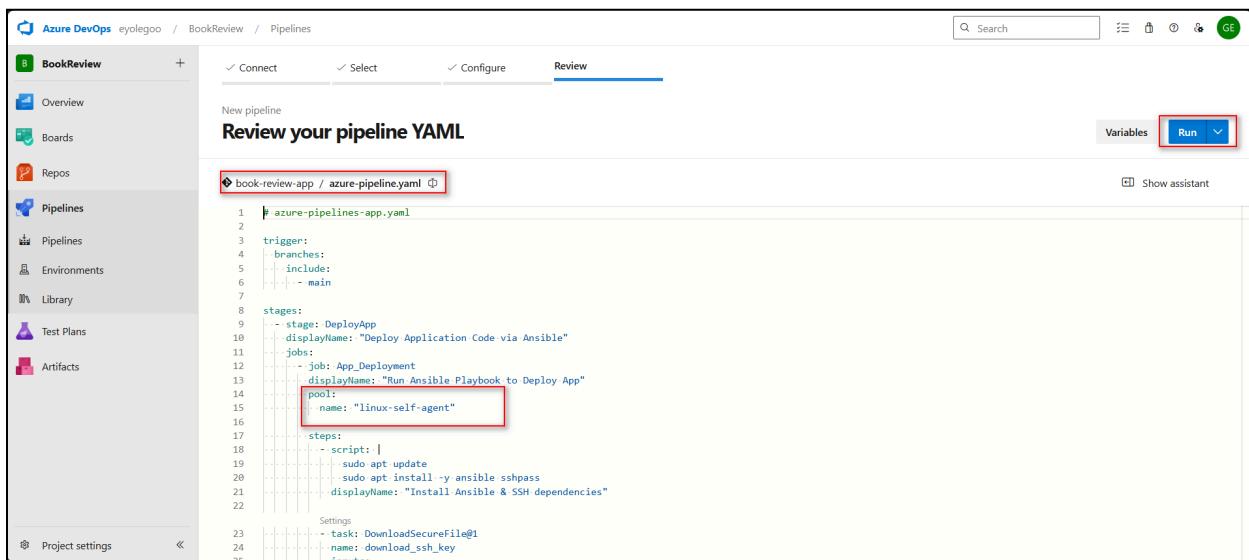
```
1 backend_repo_url:"https://github.com/pravirmishraaws/book-review-app.git"
2 backend_app_port: 3001
3 db_host: "bookreview-db1088.mysql.database.azure.com"
4 db_user: "myadmin"
5 db_password: "VerySecure0P@ssw0rd123!"
6 db_name: "book_review_db"
7 jwt_secret: "your_jwt_secret"
8 allowed_origins: "http://20.28.47.108:3000"
```



```
1 frontend_repo_url:"https://github.com/pravirmishraaws/book-review-app.git"
2 frontend_app_port: 3000
3 backend_api_url: "http://20.248.126.94:3001"
4
```

Step 5: App Repository Pipeline

1. In Azure DevOps, create a second pipeline, this time pointing to the book-review-app repo.
2. In the pipeline:
 - Install Ansible
 - Download SSH private key from Secure Files
 - Set correct permissions on the key
 - Run the main Ansible playbook that:
 - runs common role on both VMs
 - configures backend VM (app + DB connection)
 - configures frontend VM (Nginx + UI)
3. Run the pipeline and ensure all Ansible tasks succeed.



The screenshot shows the Azure DevOps Pipelines interface. On the left, there's a sidebar with project navigation: BookReview, Overview, Boards, Repos, Pipelines (selected), Pipelines, Environments, Library, Test Plans, Artifacts, and Project settings. The main area has tabs: Connect, Select, Configure, and Review (selected). Below the tabs, it says 'Review your pipeline YAML'. A file path 'book-review-app / azure-pipeline.yaml' is shown. The YAML code is listed, and the 'Run' button at the top right is highlighted with a red box.

```
# azure-pipeline-app.yaml

trigger:
branches:
include:
- main

stages:
- stage: DeployApp
  displayName: "Deploy Application Code via Ansible"
  jobs:
    - job: App_Deployment
      displayName: "Run Ansible Playbook to Deploy App"
      pool:
        name: "linux-self-agent"

      steps:
        - script: |
          sudo apt update
          sudo apt install -y ansible sshpass
        displayName: "Install Ansible & SSH dependencies"

      settings:
        - task: DownloadSecureFile@1
          name: download_ssh_key
          inputs:
```

```

- task: DownloadSecureFile@1
  name: download_ssh_key
  inputs:
    secureFile: "id_rsa"

- script: |
  mkdir -p ~/.ssh
  cp $(download_ssh_key.secureFilePath) ~/.ssh/id_rsa
  chmod 600 ~/.ssh/id_rsa
  echo "StrictHostKeyChecking no" > ~/.ssh/config
  displayName: "Setup SSH Private Key"

- script: |
  cd ansible
  ansible-playbook -i inventory.ini site.yml --ssh-extra-args "-o StrictHostKeyChecking=no -o UserKnownHostsFile=/dev/null"
  displayName: "Run Ansible Playbook"
  env:
    ANSIBLE_HOST_KEY_CHECKING: "False"

```

The screenshot shows the Azure DevOps interface for a pipeline named 'book-review-app (7)'. The left sidebar lists various project sections like Overview, Boards, Repos, Pipelines, Environments, Library, Test Plans, and Artifacts. The 'Pipelines' section is selected. A specific run titled 'Jobs in run #20251106.1 book-review-app (7)' is highlighted. The main pane displays the log for the 'Run Ansible Playbook' job. The log output is as follows:

```

75 TASK [frontend : Clone frontend repository] ****
76 changed: [20.28.47.108]
77
78 TASK [frontend : Set ownership to azureuser for frontend app] ****
79 changed: [20.28.47.108]
80
81 TASK [frontend : Install frontend dependencies] ****
82 changed: [20.28.47.108]
83
84 TASK [frontend : Create frontend environment file] ****
85 changed: [20.28.47.108]
86
87 TASK [frontend : Build frontend] ****
88 changed: [20.28.47.108]
89
90 TASK [frontend : Start frontend app using PM2] ****
91 changed: [20.28.47.108]
92
93 TASK [frontend : Save PM2 process list] ****
94 changed: [20.28.47.108]
95
96 PLAY RECAP ****
97 20.248.126.94 : ok=16 changed=11 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
98 20.28.47.108 : ok=14 changed=12 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
99
100
101 Finishing: Run Ansible Playbook

```

Now I tested the VM IP on the web browser

The screenshot shows a web browser window with the URL 'Not secure 20.28.47.108:3000'. The page title is 'Book Review'. The main content area displays three book reviews:

- The Pragmatic Programmer** by Andrew Hunt, rated 4.8/5.
- Clean Code** by Robert C. Martin, rated 4.7/5.
- JavaScript: The Good Parts** by Douglas Crockford, rated 4.5/5.

The Pragmatic Programmer

by Andrew Hunt

★ 4.8/5

Reviews

Godwin
This is a must read for everyone, great book.
★ 5/5

John Doe
Fantastic book!
★ 5/5

Add a Review
Write your review...

Created resources on azure portal

Name	Type	Location
bookreview-db1008	Azure Database for MySQL...	Australia Central
bookreview-nsg	Network security group	Australia Central
bookreview-vnet	Virtual network	Australia Central
DevOps1-pm1-dev-backend-nic	Network Interface	Australia Central
DevOps1-pm1-dev-backend-pip	Public IP address	Australia Central
DevOps1-pm1-dev-backend-vm	Virtual machine	Australia Central
DevOps1-pm1-dev-backend-vm-disk	Disk	Australia Central
DevOps1-pm1-dev-frontend-nic	Network Interface	Australia Central
DevOps1-pm1-dev-frontend-pip	Public IP address	Australia Central
DevOps1-pm1-dev-frontend-vm	Virtual machine	Australia Central

Step 5.1: Reflection

The infrastructure pipeline took the most time since setting up Terraform modules, configuring the service connection, and verifying Azure resources required careful testing. Once the infrastructure was ready, the application pipeline was smoother, though fine-tuning Ansible roles took some effort. In the next version, I'd automate the handoff by feeding Terraform outputs directly into the Ansible pipeline using artifacts or variable groups, removing manual edits entirely.