**Slide 1: Title Slide**

**Project:** Automated VM Deployment in Existing DR Infrastructure  
🚀 **Using Terraform & Azure DevOps Release Pipeline**

**Slide 2: Introduction**

We are automating the creation of Virtual Machines (VMs) within an existing Disaster Recovery (DR) infrastructure using Terraform.  
This process is integrated into an **Azure DevOps Release Pipeline** with a **manual approval stage before deployment** to ensure controlled execution.

**Slide 3: Existing DR Infrastructure**

Our DR environment includes:  
✅ **Virtual Network (VNet)**  
✅ **Subnet (SNet)**  
✅ **Network Security Group (NSG)**  
✅ **Pre-configured DR Policies and Setup**

**Slide 4: Terraform for VM Creation**

Terraform is used to provision VMs while leveraging the existing DR setup:  
✅ Uses **existing VNet & SNet**  
✅ Applies **existing NSG** for security policies  
✅ Supports **Managed Disks & Availability Sets** (if required)  
✅ Utilizes **Terraform VM Extensions** for post-creation software installation

**Slide 5: VM Extension for Software Installation**

After VM creation, Terraform VM Extensions handle software installation and configuration:

✅ **Linux VMs:**

* node\_exporter-1.9.0.linux-amd64.tar.gz (Monitoring)
* SentinelAgent\_linux\_x86\_64\_v24\_2\_2\_20.deb (Security)
* Install **nfs-common** for NFS support
* Add **additional users**
* Disable **firewalls** for necessary configurations

✅ **Windows VMs:**

* windows\_exporter-0.26.1-amd64.msi (Monitoring)
* SentinelInstaller\_windows\_64bit\_v24\_1\_5\_277.msi (Security)
* wincollect-10.1.3-24.x64\_6\_3\_(1).msi (Log Collection)

✅ **Secure credential management via Azure Key Vault:**

* **Sentinel site token**
* **Admin password**
* **User password**

**Slide 6: Azure DevOps Release Pipeline Workflow**

📌 **Trigger:** Manually executed by the user  
📌 **Approval Stage:** **Requires manual approval from Abdul Hamed** before deployment  
📌 **Terraform Execution:**  
1️⃣ **Terraform Init** – Initializes the backend state  
2️⃣ **Terraform Plan** – Previews changes  
3️⃣ **Terraform Apply** – Deploys VMs in **the existing DR infrastructure**  
📌 **Post-Deployment Tasks:**

* Install software via VM Extensions
* Configure **users, NFS, and firewall settings** on Linux VMs

**Slide 7: Pipeline Flow Diagram**

1️⃣ **User manually triggers the pipeline**  
⬇️  
2️⃣ **Approval required – Abdul Hamed**  
⬇️  
3️⃣ **Terraform deploys VMs within the DR infrastructure**  
⬇️  
4️⃣ **Software installation & configurations via VM Extension**  
⬇️  
✅ **Deployment completed successfully**

**Slide 8: Benefits of This Automation**

🚀 **Faster & more consistent VM deployments**  
🔒 **Ensures security compliance with DR policies**  
💰 **Reduces costs & saves time through automation**  
🛠️ **Seamless software installation & configuration**  
📊 **Approval-based process ensures controlled execution**

**Slide 9: Next Steps & Enhancements**

🔹 **Support for auto-scaling in future updates**  
🔹 **Integration with infrastructure monitoring tools**  
🔹 **Enhancing security policies via Terraform**

**Slide 10: Conclusion**

✅ **Automated, secure, and controlled VM deployment**  
✅ **Seamless integration with Azure DevOps**  
✅ **Approval-based deployment ensures governance & compliance**

┌──────────────────────────────────────────────┐

│ Automated VM Deployment - Flow Diagram │

└──────────────────────────────────────────────┘

[1] User manually triggers the pipeline

│

▼

[2] Approval required from Abdul Hamed

│

▼

[3] Terraform deploys VMs in DR infrastructure

│

▼

[4] Post-deployment software installation:

├── Linux VMs:

│ ├── Install node\_exporter

│ ├── Install Sentinel Agent

│ ├── Install nfs-common

│ ├── Add additional users

│ └── Disable firewall

│

├── Windows VMs:

│ ├── Install windows\_exporter

│ ├── Install Sentinel Installer

│ └── Install WinCollect

│

├── Secure credentials retrieved from Azure Key Vault:

│ ├── Sentinel site token

│ ├── Admin password

│ └── User password

│

▼

[✅] Deployment completed successfully!