**Proposed Tools and Technologies**

**1. Configuration Management & Automation**

* **Ansible**
  + **Why**: Agentless, works over SSH/WinRM, easy to integrate with existing infrastructure.
  + **Functionality**: Automate VM provisioning, configuration management, and deployment of applications.
  + **Implementation**:
    - Playbooks to define desired state.
    - Automate the configuration of VMs and underlying infrastructure across both data centers.
* **PowerShell DSC (Desired State Configuration)**
  + **Why**: Native to Windows environments, integrates well with Hyper-V.
  + **Functionality**: Ensure VMs maintain a consistent configuration state.
  + **Implementation**:
    - Use for managing and enforcing configurations on Hyper-V hosts and VMs.

**2. Testing Scenarios**

* **Packer**
  + **Why**: Automates the creation of VM images with predefined configurations.
  + **Functionality**: Build and test VM images before deployment.
  + **Implementation**:
    - Create reusable VM images for testing and production.
    - Integrate with Ansible to configure images.
* **Jenkins or Azure DevOps**
  + **Why**: Continuous Integration/Continuous Deployment (CI/CD) pipelines.
  + **Functionality**: Automate the deployment of VMs, run tests, and rollback if needed.
  + **Implementation**:
    - Set up pipelines that deploy VMs, run automated tests, and verify results.
    - Use for code-driven infrastructure as well as application deployments.

**3. Monitoring & Metric Collection**

* **Prometheus + Grafana**
  + **Why**: Prometheus is a robust metric collection tool, while Grafana provides visualization.
  + **Functionality**: Collect and display performance metrics from VMs and Hyper-V hosts.
  + **Implementation**:
    - Set up Prometheus to scrape metrics from Hyper-V hosts, VMs, and Ansible-managed endpoints.
    - Use Grafana to create dashboards for real-time monitoring.
* **Windows Performance Monitor (PerfMon)**
  + **Why**: Native tool for in-depth performance metrics in Windows environments.
  + **Functionality**: Collect detailed metrics from Hyper-V hosts and VMs.
  + **Implementation**:
    - Use alongside Prometheus for in-depth analysis of Windows-specific metrics.

**4. Alerting**

* **Nagios or Zabbix**
  + **Why**: Both are established tools with strong support for alerting.
  + **Functionality**: Set up monitoring for critical metrics and send alerts via email, SMS, or integrations with chat tools (e.g., Slack, Microsoft Teams).
  + **Implementation**:
    - Integrate with Prometheus and Ansible to ensure complete coverage of all critical systems.
    - Customize alerts based on thresholds and anomaly detection.

**5. Backup & Disaster Recovery**

* **Veeam Backup & Replication**
  + **Why**: Comprehensive backup solution for virtualized environments.
  + **Functionality**: Backup VMs and ensure quick recovery in case of failure.
  + **Implementation**:
    - Regularly back up all VMs in both data centers.
    - Automate the backup process and test recovery scenarios periodically.

**6. Cloud Integration**

* **Azure Arc**
  + **Why**: Extends Azure management capabilities to on-prem environments.
  + **Functionality**: Manage on-prem resources like they are in Azure, apply policies, and monitor via Azure Monitor.
  + **Implementation**:
    - Use Azure Arc to connect on-prem VMs and Hyper-V hosts to Azure.
    - Centralize monitoring and management via Azure.

**High-Level Architecture Diagram**

1. **Data Centers**
   * Each data center contains Hyper-V hosts with multiple VMs.
   * VMs are provisioned and configured using Ansible and PowerShell DSC.
   * Veeam manages backups.
2. **Automation & Testing**
   * Jenkins/Azure DevOps pipelines automate deployments and tests.
   * Packer builds consistent VM images.
3. **Monitoring & Alerting**
   * Prometheus collects metrics from VMs and hosts.
   * Grafana visualizes metrics.
   * Nagios/Zabbix triggers alerts on threshold breaches.
4. **Cloud Integration**
   * Azure Arc connects on-prem resources to Azure for centralized management.

**Architecture Diagram**

