Designing an automated approach to on-prem server management for a multi-data center environment involves integrating various tools and technologies to achieve the required automation, testing, reliability, metric collection, alerting, and management. Below is a comprehensive system design and the recommended tools and technologies.

System Design Overview

Environment Details

- Data Centers : 2
- Hyper-V Hosts: Several VMs
- Cloud Access: Required for management and integration

Key Components and Tools

Infrastructure Management and Automation

Hyper-V Management: Use Microsoft System Center Virtual Machine Manager (SCVMM)** for managing and optimizing the Hyper-V infrastructure.

Configuration Management: Use Ansible for configuration management and automation of tasks across the VMs and hosts.

Continuous Integration and Deployment (CI/CD)

- CI/CD Pipeline: Implement using Jenkins or Azure DevOps. These tools can automate the deployment of applications and infrastructure changes.
- Infrastructure as Code (IaC): Utilize Terraform to define and provision infrastructure in a consistent and repeatable manner.

Testing Scenarios

- Automated Testing: Integrate Selenium for end-to-end testing, JUnit or NUnit for unit testing, and Postmanfor API testing within the CI/CD pipeline.
- Test Orchestration: Use TestRail or **Xray* for managing and reporting test cases and results.

Reliability and High Availability

- Cluster Management: Use Windows Server Failover Clustering (WSFC)for high availability of Hyper-V VMs.
- Backup and Disaster Recovery: Use Veeam Backup & Replication to ensure data protection and quick recovery in case of failures.

Metric Collection and Monitoring

- Monitoring and Metrics: Deploy Prometheusfor metric collection and Grafana for visualizing metrics. Alternatively, use Azure Monitor for comprehensive monitoring.
- Log Management: Implement ELK Stack (Elasticsearch, Logstash, Kibana) for centralized logging and analysis.

Alerting

-Alerting System: Configure Alertmanager with Prometheus for alerting based on metrics thresholds. Use PagerDuty or Opsgenie for incident management and notifications.

- Notifications: Integrate with Slack, Microsoft Teams, or email for real-time alerts and notifications.

Detailed System Design

Infrastructure Management and Automation

- SCVMM will manage Hyper-V hosts and VMs across both data centers.
- Ansible will automate configuration tasks such as patching, updates, and compliance checks.

CI/CD Pipeline

- Jenkins orAzure DevOps will be used to set up pipelines for automated deployment of infrastructure (using Terraform scripts) and applications.
- Pipelines will include stages for code checkout, build, test, deploy, and rollback.

Testing

- Automated tests will be integrated into the CI/CD pipeline using tools like Selenium, JUnit, and Postman.
- Test results will be reported and managed using TestRail or Xray.

Reliability

- WSFC will ensure that VMs have high availability and can failover in case of host issues.
- Veeam will provide reliable backup and restore capabilities.

Monitoring and Metrics

- Prometheuswill scrape metrics from various components including VMs, applications, and network devices.
- Grafana will provide dashboards for real-time visualization of the collected metrics.

-Azure Monitor can offer additional insights and monitoring capabilities.

Alerting

-Alertmanager will handle alert notifications based on Prometheus metrics.

-PagerDuty or Opsgenie will manage incident responses and escalate issues to the on-call team.

- Alerts and notifications will be sent to relevant channels in Slack or Microsoft Teams.

Integration and Workflow

Provisioning and Configuration

- Terraform scripts define the infrastructure and are stored in a version control system (e.g., GitHub).

-Jenkinsor Azure DevOpstriggers infrastructure provisioning using Terraform scripts.

- Ansible playbooks configure the VMs and install necessary software.

Continuous Deployment

- Application code is committed to a repository and triggers the CI/CD pipeline.
- The pipeline runs automated tests, builds the application, and deploys it to the VMs.
- Post-deployment tests ensure the application is running correctly.

Monitoring and Alerting

- Prometheus collects metrics from VMs and applications.
- Grafana dashboards display real-time metrics and health status.
- Alertmanager triggers alerts based on predefined thresholds and sends notifications to the incident management system.
- On-call engineers receive alerts via PagerDuty or Opsgenie and respond to incidents.

Conclusion

This automated approach leverages a combination of open-source tools and managed services to provide a robust, scalable, and reliable server management solution. The integration of CI/CD pipelines ensures continuous delivery and testing, while comprehensive monitoring and alerting systems ensure high availability and quick incident response.