Monitoring disk usage is critical to maintaining a healthy infrastructure. When disks approach full capacity, applications can fail, logs may be lost, and services can become unresponsive. By automating disk usage checks with Ansible, you can periodically gather usage data across many servers and trigger alerts if usage exceeds defined thresholds. This approach allows you to proactively address capacity issues and prevent downtime or data loss.

**Access management-**

**Manage access to all VM’s in subscription:**

* **Azure Credentials:**

Ensure you have appropriate Azure credentials configured for Ansible to interact with your resources. This typically involves setting up a service principal or using Azure CLI authentication.

**Use RBAC roles to define access on the Azure resources-** <https://learn.microsoft.com/en-us/azure/role-based-access-control/built-in-roles>

* Assign least-privilege roles to the Ansible control node:
  + For Linux VMs: Virtual Machine Administrator Login
  + For Windows VMs: Virtual Machine User Login
* Use Managed Identities or Service Principals with RBAC to authenticate securely

**Secure Credential Storage-**

* Use Ansible Vault to encrypt sensitive data like SSH keys, WinRM credentials, or Azure client secrets.
* Rotate credentials periodically and enforce strong password policies.

**Ansible collect Data from other VM’s-**

**1. Dynamic Inventory with Azure Plugin**

* Use the azure\_rm plugin in Ansible to dynamically discover VMs across subscriptions.
* Filter VMs using tags like MonitoringEnabled=true.

**2. Connection Methods**

* **Linux VMs**: Connect via SSH using key-based authentication.
* **Windows VMs**: Connect via WinRM over HTTPS with encrypted credentials.

**3. Network Security**

* Use **Azure Bastion** or **Jump Hosts** to access VMs securely without exposing them to the public internet.
* Ensure **NSGs (Network Security Groups)** allow only necessary traffic from the Ansible control node.

**4. Resilience and Reliability**

* Implement **retry logic** in playbooks to handle transient failures.
* Use **parallel execution** in Ansible to speed up data collection across many VMs.

**VM Discovery and Enrollment for metric collection**

**Gather Disk Usage Data-**

**To gather disk usage data from all Azure VMs and centralize it for monitoring, we’ll use a combination of Ansible, Azure-native services, and secure data aggregation techniques.**

**1. VM Discovery**

* Use Ansible’s azure\_rm dynamic inventory plugin to discover VMs across subscriptions.
* Filter VMs using tags like MonitoringEnabled=true to include only relevant ones.

**2. Disk Usage Collection via Ansible**

* Linux VMs: Use SSH to run df -h or lsblk to get disk usage.
* Windows VMs: Use WinRM to run Get-PSDrive or Get-WmiObject Win32\_LogicalDisk.

**3. Sample Ansible Tasks**

**YAML**

**- name: Collect disk usage**

**hosts: all**

**gather\_facts: no**

**tasks:**

**- name: Get disk usage on Linux**

**shell: df -h**

**register: disk\_output**

**when: ansible\_os\_family != "Windows"**

**- name: Get disk usage on Windows**

**win\_shell: Get-PSDrive**

**register: disk\_output**

**when: ansible\_os\_family == "Windows"**

**- name: Save output to local file**

**copy:**

**content: "{{ disk\_output.stdout }}"**

**dest: "/var/log/disk\_usage/{{ inventory\_hostname }}.log"**

**Centralizing and Presenting the Data**

**Option 1: Azure Log Analytics (Recommended)**

* Use Azure Monitor Agent or custom ingestion via REST API to push Ansible-collected logs to Log Analytics Workspace.
* Create Kusto queries to visualize disk usage.
* Set up Alerts for thresholds (e.g., disk usage > 80%).

**Option 2: Custom Dashboard**

* Push logs to a central Azure Storage Account or Blob Container.
* Use Power BI or Grafana (connected to Azure Monitor or Blob Storage) to visualize disk usage trends.

**Option 3: Email or Teams Notifications**

* Use Ansible to send alerts via email or webhook to Microsoft Teams when disk usage exceeds a threshold.
* Use tags like MonitoringEnabled=true to identify VMs that should be monitored.
* As new VMs are created, applying this tag automatically includes them in monitoring

**VM Growth Management**

**Dynamic Inventory with Ansible**

* Use the azure\_rm plugin to dynamically discover VMs across all delegated subscriptions.
* Automatically includes new VMs without manual updates to inventory files.

**Scheduled Execution**

* Use cron jobs or Azure Automation to run Ansible playbooks periodically.
* Ensures disk usage data is collected regularly, even as VM count increases.

**Data Aggregation and Storage**

**Centralized Logging**

* Push disk usage data to a Log Analytics Workspace or Azure Storage Account.
* These services scale automatically with data volume.

**Query and Visualization**

* Use Kusto queries in Log Analytics or Power BI dashboards to visualize trends.
* Dashboards can be configured to auto-refresh and include new data sources.

**Alerting and Response**

**Azure Monitor Alerts**

* Set up alerts based on disk usage thresholds.
* Alerts can be configured to apply to all VMs in a workspace, including newly added ones.

**Requirements**

* Ansible (on the control machine) version 2.9 or higher
* SSH access (root or a user with sudo privileges) on each target server
* Python 2.7+ or 3.x installed on each target server (for Ansible modules)
* Basic command-line utilities (e.g., df, awk) available on the target servers
* A method for sending alerts (e.g., an SMTP server for email, or a webhook/CLI tool for chat notifications)

Kindly refer to this document if you want to configure Ansibel on Azure VM: <https://learn.microsoft.com/en-us/azure/developer/ansible/install-on-linux-vm?tabs=azure-cli>

To monitor disk utilization on Azure VMs using Ansible, you can leverage the azure\_rm\_virtualmachine\_info module to gather disk information and then use Ansible's assert module or other logic to check for threshold violations and trigger alerts. Here's a breakdown of the process:

1. Inventory and Azure Credentials:

* **Inventory:**

Create an Ansible inventory file (hosts.ini or similar) that lists your Azure VMs, including their connection details (e.g., ansible\_host, ansible\_user, ansible\_ssh\_private\_key\_file).

* **Azure Credentials:**

Ensure you have appropriate Azure credentials configured for Ansible to interact with your resources. This typically involves setting up a service principal or using Azure CLI authentication.

2. Ansible Playbook:

Code

---  
- name: Monitor Azure VM Disk Utilization  
 hosts: your\_azure\_vms *# Replace with your inventory group*  
 gather\_facts: false  
 tasks:  
 - name: Gather disk information  
 azure.azcollection.azure\_rm\_virtualmachine\_info:  
 resource\_group: your\_resource\_group *# Replace with your resource group*  
 name: "{{ item }}"  
 *# other parameters, such as location*  
 register: vm\_info  
 loop: "{{ groups['your\_azure\_vms'] }}" *# Or a list of specific VM names*  
  
 - name: Check disk utilization  
 block:  
 - name: Get total and free disk space  
 set\_fact:  
 disk\_data: "{{ item.storage\_profile.os\_disk.managed\_disk.disk\_size\_gb | int \* 1024 \* 1024 \* 1024 | float, item.storage\_profile.os\_disk.managed\_disk.disk\_size\_gb | int \* 1024 \* 1024 \* 1024 | float}}" *# get disk size from item and calculate bytes from GB*  
 loop: "{{ vm\_info.virtualmachines }}"  
 loop\_control:  
 label: "{{ item.name }}"  
  
 - name: Calculate utilization percentage  
 set\_fact:  
 disk\_utilization: "{{ ( (disk\_data.0 - disk\_data.1) / disk\_data.0) \* 100 }}"  
  
  
 - name: Trigger alert if utilization is high  
 fail:  
 msg: "High disk utilization on VM {{ item.name }}. Utilization: {{ disk\_utilization | round(2) }}%"  
 when: disk\_utilization > 80 *# Example threshold*  
  
  
 - name: "Display disk usage information for {{ item.name }}"  
 debug:  
 msg: "Disk usage on {{ item.name }}: {{ disk\_utilization | round(2) }}%"

3.  Explanation:

* azure\_rm\_virtualmachine\_info:

This module retrieves detailed information about your Azure VMs, including their operating system and data disks.

* set\_fact:

Used to calculate the disk utilization percentage. It first gets the total and free space (as bytes from GB), then computes the utilization.

* fail:

If the utilization exceeds a threshold (e.g., 80%), the fail module will stop the playbook and print an error message.

* when:

The when condition ensures that the fail task is only executed when the disk utilization is above the set threshold.

* debug:

This task displays the disk usage information for each VM, making it easier to monitor the results.

* loop:

The tasks are looped over each VM in your inventory group.

4. Running the Playbook:

* Save the playbook as a YAML file (e.g., monitor\_disks.yml).
* Execute it using ansible-playbook monitor\_disks.yml -i hosts.ini (replace hosts.ini with your inventory file).