oVirt - Windows Template

To create an Ansible playbook that deploys a Windows VM from a template on oVirt, customizes the OS with sysprep, sets unique hostname and static IP, and performs other specified configurations, follow the structure below. This example assumes you have a sysprep file ready for Windows customization and your oVirt environment is properly set up for Ansible integration.

First, ensure you have the `ovirt.ovirt` collection installed, which includes modules for interacting with oVirt. If not, you can install it using Ansible Galaxy:

```
ansible-galaxy collection install ovirt.ovirt
```

Here's an example playbook that meets your requirements. You'll need to adjust variables and possibly the paths to files (like the sysprep file) to match your environment.

```
---
- name: Deploy and customize a Windows VM on oVirt
  hosts: localhost
  gather_facts: no
  collections:
    - ovirt.ovirt

  vars:
    ovirt_engine_url: https://ovirt-engine.example.com/ovirt-engine/api
    ovirt_engine_username: admin@internal
    ovirt_engine_password: your_password
    ovirt_engine_cafile: /path/to/your/ovirt-engine.ca
    vm_domain: "example.com"
    vm_subnet: "255.255.255.0"
    vm_gateway: "10.10.10.1"
    vm_dns: "10.1.10.10"
    additional_disk_size: 20GB
    machines:
      - { name: dc01, memory: 4GiB, cluster: kvm_worker, template: Windows2022Core, datasize: 90, storage: data-kvm2, tag: lab, ip: 10.10.10.12}
      - { name: dc02, memory: 4GiB, cluster: kvm_worker, template: Windows2022Core, datasize: 90, storage: data-kvm2, tag: lab, ip: 10.10.10.11}
      - { name: wadm01, memory: 8GiB, cluster: kvm_worker,  template: Windows2022, datasize: 90, storage:
```

```yaml
data-kvm2, tag: lab, ip: 10.10.10.10}

  tasks:
    - name: Log into oVirt
      ovirt.ovirt.ovirt_auth:
        url: "{{ ovirt_engine_url }}"
        username: "{{ ovirt_engine_username }}"
        password: "{{ ovirt_engine_password }}"
        ca_file: "{{ ovirt_engine_cafile }}"
        state: present

    - name: Deploy VMs
      ovirt.ovirt.ovirt_vm:
        auth: "{{ ovirt_auth }}"
        name: "{{ item.name }}.{{ vm_domain }}"
        template: "{{ item.template }}"
        cluster: "{{ item.cluster }}"
        cpu_cores: 2
        cpu_sockets: 1
        memory: "{{ item.memory }}"
        sysprep:
          hostname: "{{ item.name | upper}}"
          ip: "{{ item.ip }}"
          netmask: "{{ vm_subnet }}"
          gateway: "{{ vm_gateway }}"
          dns_servers: "{{ vm_dns }}"
          domain: "{{ vm_domain }}"
          root_password: "{{ vm_admin }}"
        state: present
      with_items:
        - "{{ machines }}"

    - name: Add Software Storage
      ovirt.ovirt.ovirt_disk:
        auth: "{{ ovirt_auth }}"
        name: "{{ item.name }}-Disk2"
        vm_name: "{{ item.name }}.{{ vm_domain }}"
        size: "{{ item.datasize }}GiB"
        format: cow
        interface: virtio_scsi
```

```yaml
        storage_domain: "{{ item.storage }}"
      with_items:
        - "{{ machines }}"


    - name: Start VMs
      ovirt.ovirt.ovirt_vm:
        auth: "{{ ovirt_auth }}"
        name: "{{ item.name }}.{{ vm_domain }}"
        state: running
      with_items:
        - "{{ machines }}"


    - name: Tag machines
      ovirt.ovirt.ovirt_tag:
        auth: "{{ ovirt_auth }}"
        name: "{{ item.tag }}"
        state: attached
        vms:
          - "{{ item.name }}.{{ vm_domain }}"
      with_items:
        - "{{ machines }}"



    # Assuming the VM is to be powered on after setup
    - name: VMs should be running
      ovirt.ovirt.ovirt_vm:
        auth: "{{ ovirt_auth }}"
        name: "{{ vm_hostname }}"
        state: running


    - name: Logout from oVirt
      ovirt.ovirt.ovirt_auth:
        state: absent
        auth: "{{ ovirt_auth }}"
```

Remember to replace placeholders (like URLs, credentials, paths, domain names, and the storage domain) with your actual data. Also, ensure your sysprep file is correctly set up in your template or specified directly in the playbook if needed.

This playbook performs the following actions:

1. Logs into the oVirt engine.
2. Creates a VM from a specified template with a unique hostname and configures it with sysprep.
3. Adds an additional 100GB disk to the VM.
4. Configures the VM's network interface.
5. Powers on the VM after setup.
6. Logs out from the oVirt engine.

Test this playbook in a development environment before using it in production. Adjustments may be necessary based on your specific oVirt setup, Windows template, and network configuration.