Accelerating DevOps with HPE OneView and Ansible
Executive summary

As information technology becomes an integral part of products, services, and experiences, a new style of infrastructure is emerging where resources can be composed, decomposed, and recomposed from a fluid pool of compute, storage, and fabric infrastructure. The Hewlett Packard Enterprise vision of Composable Infrastructure brings together software-defined resources with template-driven provisioning and management to provide the right resource at the right time for each workload in the data center. Composability brings the public cloud experience to your business using the hardware inside your own data center.

To make this vision a reality, Hewlett Packard Enterprise has teamed up with several industry-leading configuration management providers, including Ansible by Red Hat®. The Ansible tool gives developers fast, scalable, and flexible automation of application configuration, deployment, and orchestration. The integration of Ansible Core with HPE OneView extends giving DevOps teams the ability to automate the provisioning of bare-metal resources, including servers, storage, and networking as part of the application deployment process. This accelerates time to value through automated, consistent provisioning.

This white paper shows IT architects and administrators how to use software-defined templates in HPE OneView to automate the provisioning of physical infrastructure in conjunction with Ansible Core. HPE continually collaborates with Red Hat to gain an understanding on how this current integration can be enhanced to operate seamlessly with Ansible’s Enterprise Class family of Orchestration and Management tools.

Business context: Composable Infrastructure

Today, lines of business are putting tremendous pressure on IT leadership in this Idea Economy to enable new applications and services such as mobile apps, social media, cloud, and the Internet of Things (IoT). These applications and services are location-aware and drive real-time, personalized customer experiences and revenue opportunities. At the same time, IT must still manage the traditional applications that run the business—enterprise resource planning (ERP), email, databases, and more.

The challenge is that these two different roles require two distinctly different approaches. In traditional IT, change is infrequent; infrastructure is static and applications spin a few times a year at most. With new mobile and cloud-native applications, change delivery is continuous, with many companies spinning mobile and cloud-native applications daily or even more frequently. To accomplish this, DevOps teams need dynamic infrastructure that can be deployed on demand.

While the public cloud can provide an on-demand option for some new applications, it can also present IT with enterprise architecture concerns over issues such as security and data compliance. CIOs that want to host applications internally face the need for two separate infrastructures, one designed for reliability and stability to support mature applications and a second designed for agility and speed to support mobile and cloud-native applications. This is costly and complex, and is inconsistent with another directive that most CIOs confront—to simplify operations and reduce the overall cost of IT.

To address these competing needs, Composable Infrastructure delivers on both models. The concept is simple—physical and virtual resources can be provisioned to applications on-demand regardless of the style of application or the type or quantity of the resource.
Composable Infrastructure is built on three basic architectural principles:

- **Fluid resource pools**—fluid pools of compute, storage, and fabric resources that are boot-up ready for any physical or virtual workload and are composed instantly to support the needs of specific applications and services.

- **Software-defined intelligence**—template-driven workload automation to implement changes quickly and programmatically for frictionless operations.

- **A unified API**—a single open API that provides a single line of code to abstract every element delivering a true infrastructure-as-a-code service.

Think of a Composable Infrastructure as a set of flexible building blocks that can be dynamically and automatically assembled and re-assembled to meet changing workload needs. Composable Infrastructure has the unique ability to run without being limited to one computing paradigm. It can support traditional and cloud-native apps in virtual machines, containers, or on bare metal. This infrastructure can run and store anything. It also fully integrates with industry-standard tools such as Ansible, SaltStack, Chef, Docker, VMware®, OpenStack®, and others, allowing IT operators and developers to continue leveraging the tools they use every day.

To learn more, read the white paper “HPE Composable Infrastructure: Bridging traditional IT with the Idea Economy.”

---

**Figure 1:** Composable Infrastructure principles
Provisioning with HPE OneView

To create an agile and efficient IT infrastructure, organizations must transform traditional rigid physical systems into flexible physical or virtual resource pools. As they do this, they are often challenged to automate the provisioning of the physical, bare-metal hardware. Provisioning is a complex process, requiring configuration of hardware, connectivity, OS, and application stacks on the right resources with the right configurations. A typical provisioning process might include these steps:

- Update firmware
- Update drivers
- Set BIOS settings
- Set unique identifiers (WWN, SN, UUID, MAC)
- Install OS
- Configure Smart Array
- Configure network connectivity
- Configure SAN zoning
- Configure SAN array

Using traditional processes, standing up infrastructure can require three or four independent teams (compute, storage, networking, and more). These teams use several different tools to execute more than 50 separate scripts consisting of thousands of lines of code, and requiring manual intervention and monitoring along the way. It can take days or weeks to integrate and stand up a new server.

Composable Infrastructure allows IT to create re-useable profile templates to define infrastructure configuration and settings across servers, storage, and networking, which can then be provisioned from bare-metal resources through a single line of code. Templates created in HPE OneView allow IT to create logical infrastructures from available resource pools, specific to each application’s needs.

HPE OneView utilizes a single, unified composable API to provide a consistent interface for discovering, searching, inventorying, configuring, provisioning, updating, and diagnosing composable resources. Using rich, high-level infrastructure constructs, it eliminates the need for multiple, time-consuming, low-level programming interfaces so administrators and software developers can be more productive.

In addition to initial provisioning, change operations such as adding additional storage to a service, modifying network connectivity, or updating firmware can also be implemented via templates. The use of repeatable templates allows changes to be deployed automatically and significantly reduces manual interaction and errors.

In short, instead of using dozens of scripts to automate physical infrastructure, Composable Infrastructure lets IT admins and DevOps teams use just a single line of code to orchestrate all of the steps needed to set up a physical infrastructure. The unified API makes it possible to treat physical infrastructure as code with infrastructure management software.
Accelerating DevOps with HPE OneView and Ansible

Ansible is an open source community project sponsored by Red Hat, providing a simple IT automation engine that automates cloud provisioning, configuration management, application deployment, intra-service orchestration, and many other IT needs.

Ansible works by connecting to server nodes and pushing out small programs, called “Ansible Modules” to those nodes. Programs are written to be resource models of the desired state of the system. Ansible executes these modules (over SSH by default), and removes them when finished. A library of modules can reside on any machine, and there are no servers, daemons, or databases required. Typically, you’ll work with your favorite terminal program, a text editor, and a version control system to keep track of changes to your content.

Ansible uses YAML, a simple, human readable markup language in playbooks to automate and orchestrate the build, deployment, and management of an application’s software stack.

Ansible playbooks can be version-controlled and tested just like application software, providing repeatable and reliable installations and upgrades.

Using this simple Ansible playbook, DevOps can automate a task such as creation of a web server cluster with a load balancer:

```
# Configure and deploy the web servers.
- hosts: webservers
  remote_user: root
  roles:
    - base-apache
    - web

# Configure and deploy the load balancer.
- hosts: lbservers
  remote_user: root
  roles:
    - haproxy
```

This playbook assumes that servers are ready and waiting to land the application stack with hardware configured and the OS installed. Provisioning that hardware stack is a multi-step process, requiring many tools to manage provisioning tasks. Provisioning infrastructure can easily become a bottleneck in the continuous delivery of applications.

Traditional process—needs a different tool for every task

Figure 2: Traditional multi-step process for provisioning hardware is inefficient
HPE OneView manages all these provisioning functions through a single API, leveraging pre-existing profiles and templates. Binding Ansible with HPE OneView allows DevOps to introduce physical provisioning into the same playbook used to deploy the software stack. Adding an additional line of code to our Ansible playbook directs HPE OneView to provision hardware and to load the operating system using specified templates.

```
# Deploy physical servers with an OS
- hosts: all-servers
  gather_facts: no
  roles:
    - hp-oneview-server

# Configure and deploy the web servers.
- hosts: webservers
  remote_user: root
  roles:
    - base-apache
    - web

# Configure and deploy the load balancer.
- hosts: lb servers
  remote_user: root
  roles:
    - haproxy
```

This playbook calls the Ansible role for HPE OneView to physically provision servers from bare metal and configures networks, storage, BIOS, and firmware. It also uses the OS provisioning module of HPE OneView—Insight Control server provisioning (ICsp) to install an OS. The Ansible playbook then configures the application stack and assigns the servers to a load balancer, all in a single flow. Using Ansible with HPE OneView also allows automated non-disruptive upgrades from the physical bare metal all the way up through the software stack.
HPE OneView and Ansible provide a software-defined approach to the management of the entire hardware and software stack, giving IT the ability to deliver new or updated services on an as-needed or on-demand basis. The HPE OneView and Ansible integration enables provisioning on any infrastructure supported by HPE OneView, including HPE BladeSystem, HPE ProLiant servers, HPE 3PAR Storage, as well as the new HPE Synergy platform.

Benefits for your business include:

- **Faster time to value:** Developers see faster build times, meaning less downtime and more productivity. Entire application stacks can be provisioned automatically from bare metal through application in minutes.

- **Increased reliability:** After integration with HPE OneView bare-metal servers are configured the same way every time, and maintain infrastructure compliance with automated rolling upgrades.

- **Deployment flexibility:** Provision and update bare-metal resources with a single line of code, in the same way as virtual and cloud resources.

The Ansible role for HPE OneView is available for download at [github.com/HewlettPackard/oneview-ansible](https://github.com/HewlettPackard/oneview-ansible).
**Walk-through**

This walk-through demonstrates provisioning of a pair of Apache web servers with load balancing from bare metal, deployment of the operating system, and deployment of a web app on those servers.

The walk-through assumes that:

- An HPE OneView appliance is present with browser access to the HPE OneView UI. ICsp is configured to deploy the operating system of choice.
- Server profile templates have been created in HPE OneView to provision servers and OS build plans, which have been created in ICsp.
- There is an available server enclosure and blade servers.

**Server profile templates**

The server profile template describes how the server will run in the data center, including firmware versions and settings, local and SAN storage configurations, boot settings, BIOS settings, network connections, and more. These templates allow you to capture the best practices for different types of workloads in a repeatable format.

Server profile templates are managed in HPE OneView. Select a server profile template to include in your Ansible playbook. In this case, we'll be using the web server template.
Executing an Ansible playbook to provision an application

Execute an Ansible playbook to provision hardware, deploy an operating system, and deploy an application stack. In this case, we'll be deploying two Linux® servers, a load balancer, and web services.

```
$ ansible-playbook -i demo/hosts ov_site.yml
```

The HPE OneView server role will create server profiles for all the servers in the stack—web servers and a load balancer. Servers are configured as defined in the template and powered on.

```
$ ansible-playbook -i demo/hosts ov_site.yml
PLAY [all-servers] *****************************************************************
TASK: [hp-oneview-server | Create Server Profiles] *******************************
changed: [demo-web2 -> localhost]
changed: [demo-web1 -> localhost]
changed: [demo-lb -> localhost]
TASK: [hp-oneview-server | Power on servers] **************************************
changed: [demo-lb -> localhost]
changed: [demo-web2 -> localhost]
changed: [demo-web1 -> localhost]
```

Progress can be monitored on the console or in HPE OneView.
Notice at the bottom left that three servers have been created and connected.

Next, the HPE OneView server role deploys the operating system on each server using ICsp.

When the OS is successfully deployed, the Ansible playbook will deploy the application stack on the web servers.

Next, the Ansible playbook will install and configure the load balancer.
The application is now deployed on two web servers behind a load balancer.

We have used a simple addition to an Ansible playbook to provision a physical infrastructure completely including loading firmware, configuring a BIOS, creating network and storage connections, and installing an OS. This collapses hundreds of lines of code into just a few, letting admins configure physical infrastructure using the code in the same way as they might configure cloud-based infrastructure.

**Executing an Ansible playbook to automate rolling updates**

Ansible playbooks support automated, non-disruptive rolling updates to the software stack. Adding an HPE Synergy Composer role to your playbook allows you to automate non-disruptive infrastructure updates, as well as software stack updates. A sample playbook with the HPE OneView role included is shown here.
In HPE OneView, update the server profile template to capture the new server configuration—firmware, BIOS, connections, and more.

HPE OneView automatically detects servers that are not consistent with the new profile settings.
Run the Ansible playbook that includes the HPE OneView role to perform a rolling update of the infrastructure and the applications.

The first server will be disabled from the load balancer and the server will be powered off. The new profile will be applied and the server will be powered back on.

```
$ansible-playbook -i demo/hosts rolling_update.yml
PLAY [webserver1] ******************************************
GATHERING FACTS *********************************************
ok: [demo-web1]
TASK: [disable the server in haproxy] *************************
changed: [demo-web1 = demo-lb] => (item=demo-lb)
TASK: [hp-oneview-server | Create Server Profiles] **********
ok: [demo-web1 => localhost]
TASK: [hp-oneview-server | Power on server] ****************
skipping: [demo-web1]
TASK: [hp-oneview-server | Deploy OS] **********************
skipping: [demo-web1]
TASK: [hp-oneview-server | Power off server to fix offline remediation] *****
changed: [demo-web1 = localhost]
TASK: [hp-oneview-server | Remediate compliance issues] ********
changed: [demo-web1 = localhost]
TASK: [hp-oneview-server | Power on server] ****************
```

Software updates are then applied to web server 1. When updates are completed on web server 1, it is re-enabled in the load balancer.

```
TASK: [hp-oneview-server | Power on server] ****************
changed: [demo-web1 = localhost]
TASK: [hp-oneview-server | Wait for server to boot] ********
ok: [demo-web1 = localhost]
TASK: [base-apache | Install http and php etc] *************
ok: [demo-web1 = item=httpd]
TASK: [base-apache | http service state] ******************
ok: [demo-web1]
TASK: [web | Install php and git] **************************
ok: [demo-web1 = item=php,php-mysql]
TASK: [web | Restart httpd] *******************************
changed: [demo-web1]
TASK: [web | Copy web-page] *******************************
ok: [demo-web1]
TASK: [wait for webservers to come up] *******************
ok: [demo-web1]
TASK: [enable the server in haproxy] *********************
changed: [demo-web1 = demo-lb] => (item=demo-lb)
TASK: [disable the server in haproxy] *********************
```
The process is then repeated for web server 2.

We’ve just shown how physical hardware can be updated non-disruptively using the same approach as used to update application code in a load balanced environment.

**Understand the provisioning**

The HPE OneView driver performs the following steps during the provisioning process:

1. **Locates and allocates compatible server hardware** from the available pool of servers managed by HPE OneView, using the HPE OneView RESTful API search.

2. **Creates new instances from the selected server profile template** using the HPE OneView RESTful API:
   a. **Firmware, BIOS, and local storage configuration**: HPE OneView uses the embedded HPE Intelligent Provisioning (IP), a Linux pre-boot environment embedded on the HPE iLO baseboard management card (BMC) processor, and the embedded SmartStart tooling.
   b. **Network connectivity**: HPE OneView uses the HPE Virtual Connect networking platform to provision network connectivity for servers dynamically, from a pool of network connections.
   c. **SAN storage**: HPE OneView automates SAN storage provisioning for HPE 3PAR SAN storage systems and Fibre Channel switches including Brocade, Cisco, and HPE Networking.

3. **Deploys the selected OS** once the hardware provisioning is complete:
   a. The OS deployment network connection in the server profile is configured to PXE boot and to the OS deployment server.
   b. The server hardware is powered on and PXE boots.
   c. The server registers with the OS deployment server.
   d. The HPE OneView driver detects the server is registered and kicks off the OS installation using the ICsp.

4. **Configures the host networking**
   a. IP address, subnet, hostname, and others for different network adapters and virtual network adapters, allowing you to build sophisticated private networking arrangements.

5. **Deploys Ansible application stack on top of the provisioned server OS**
   a. Apply the playbook commands to configure the web server.
Best practices

Here are some recommended best practices for integrating HPE OneView and Ansible to provision and deploy application workloads.

Secure your systems
Change the generic OS build plan in HPE OneView ICsp to include steps for customization to OS provisioning to secure your systems for production use. Include the following steps in your OS build plan:

- Update default passwords for root
- Change SSH access for direct root login
- Update YAML repositories to trusted repositories
- Install Ansible Certification Authority (CA) certificates for private registries for self-signed certificates

Employ configuration management best practices
Create configuration management using software configuration management (SCM) or another secure storage repository. This allows you to share the resources your developers use as a team and collaborate on those resources in a version-controlled manner.

Divide your work
When creating Ansible playbooks, maintain small playbooks that serve specific purposes, allowing re-use. For example, create a playbook that defines how to start Apache that can then be re-used through a simple "include" statement by any playbook that needs to start Apache.

Do not hard-code
Rather than hard-coding parameters, code your Ansible playbooks to be generic and put your parameters into an environment definition, or an attribute file. Then, when running the playbook, you can simply make changes to one of these resources rather than needing to re-code your playbook.

Do not reinvent the wheel
Ansible is an open source tool, built from the ground up for application and infrastructure automation, ranging from individual servers to huge, globally dispersed data centers. With thousands of users and a global community sharing best practices and source code, it is quite possible that the code you need already exists.

In addition to coding reusable playbooks as defined earlier, the Ansible Galaxy contains thousands of pre-packaged roles that can be dropped into your Ansible playbook and immediately applied to your infrastructure. You can also download roles from other sources such as GitHub and create your own YAML files that can then be added to your playbooks.

Provision and de-provision through playbooks
Your templates and profiles work together with Ansible playbooks for provisioning and for ongoing management and configuration. Remember, however, that shutting down a server to be returned to the pool is not just a matter of turning it off. Instead, you’ll want to create a playbook that does the de-provisioning as well.

Get help with your Ansible deployments
The HPE Datacenter Care—Infrastructure Automation team is staffed with experts in infrastructure-as-a-code and agile processes and can assist with integrating Ansible and HPE OneView in customer environments and with Ansible deployments.
Summary

Today's Idea Economy is driving IT leaders to find new and innovative ways to deliver the flexibility of hybrid IT solutions while reducing complexity and operating costs. Composable Infrastructure provides the promise of allowing IT to provision and manage traditional workloads along with new mobile and cloud-native applications from a single infrastructure, allowing you to achieve the vision of infrastructure as a code.

A key tool in providing infrastructure as a code is Ansible, which provides programmers, development teams, and operations engineers the common toolbox they need to take advantage of the distributed and networked nature of modern applications. HPE OneView, together with Ansible, gives your DevOps teams true infrastructure-as-a-code capabilities within your data center, supporting faster time to value for new revenue-generating applications.

Learn more at
hpe.com/info/composableprogram
hpe.com/info/oneview
ansible.com/get-started
github.com/HewlettPackard/oneview-ansible