Book of Cloud Native Services - Nutanix Kubernetes Engine (formerly Nutanix Karbon)

[PDF generated May 09 2022. For all recent updates please see the Nutanix Bible releases notes located at https://nutanixbible.com/release_notes.html. Disclaimer: Downloaded PDFs may not always contain the latest information.]

Nutanix Kubernetes Engine (NKE) is Nutanix's certified enterprise Kubernetes management solution that enables turnkey provisioning, operations, and lifecycle management of Kubernetes.

Supported Configurations

The solution is applicable to the configurations below:

Core Use Case(s):

- Containers
- Microservices
- Application modernization

Management interfaces(s):

- NKE console in Prism Central (PC)

Supported Environment(s):

- Hypervisors:
  - AHV

- Locations:
  - On-premises:
  - Owned
  - (Managed) Service Providers

Supported node OS image(s):

- CentOS Linux-based provided by Nutanix

Upgrades:

- Included in LCM as Karbon

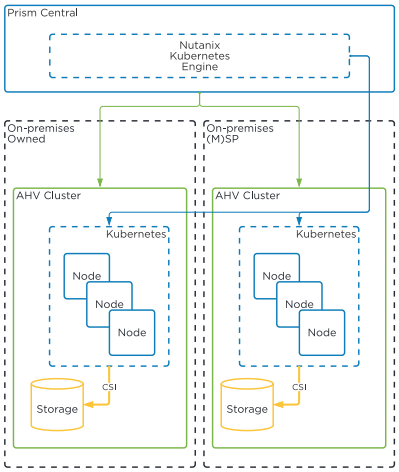
Compatible Features:

- Lifecycle operations
- Role-based Access Control (RBAC)
- Cluster expansion
- Multi-cluster management
- Node pool
- GPU pass-through

Nutanix Kubernetes Engine can be enabled via Prism Central. Any Nutanix AOS cluster registered with an NKE-enabled PC can be used as a target for provisioning Kubernetes clusters.

Architecture

An NKE-enabled Kubernetes cluster cannot span over multiple Nutanix HCI clusters



NKE Architecture

NKE runs as a containerized service in Prism Central. When NKE is enabled on a PC, two containers are provisioned under the covers: the *karbon-core* container and the *karbon-ui* container.

- *Karbon-core* is responsible for the lifecycle of Kubernetes clusters. Tasks such as provisioning, node OS upgrades, cluster expansion, and others, are performed by this container.
- *Karbon-ui* is responsible for providing an intuitive console via Prism Central. From this integrated UI the IT admins have full control over the entire Kubernetes landscape managed by the NKE instance.

Air-gapped environments

NKE can be enabled in air-gapped environments too (read more)

Kubernetes Cluster Configurations

OS Images

NKE provides the OS image for installing and scaling Kubernetes nodes. NKE uses the CentOS Linux-based operating system for NKE-enabled Kubernetes cluster creation. New OS image versions are periodically released including patches to fix vulnerabilities. For a list of supported OS image versions, check the NKE release notes (read more)

Operating System Images

Bringing your own OS image is not supported.

Compute

The recommended configurations include two options: *development* cluster and *production* cluster.

- The *development* cluster option does not have a highly available control plane. In the event of a control plane node going offline, the Kubernetes cluster will be impacted.

  The minimum cluster size for development is three nodes:

  - The control plane is divided into two nodes, the etcd node, and the Kubernetes control plane node.
  - The worker node pool has a single node that can be scaled out up to 100 nodes (NKE Configuration Maximums).

- The *production* cluster option does have a highly available control plane. There is no single-point-of-failure with this configuration option.

  The minimum cluster size for production is eight nodes:

  - The control plane is divided into five nodes, three etcd nodes - can scale up to five nodes -, and two Kubernetes control plane nodes. The Kubernetes control plane nodes can operate as active-passive (two nodes), or active-active (up to five nodes). For the latter, an external load balancer is required.
  - The worker node pool has a minimum of three nodes that can be scaled out up to 100 nodes.

Networking

There are a total of three networks required by a Kubernetes cluster which can be grouped into virtual machines network and Kubernetes networks.

- Virtual machines network or node network. This has to be allocated either by DHCP (development clusters only) or via a Managed network that is IPAM enabled (with associated domain settings and IP address pools). Production configuration requires additional static IP addresses for active-passive, and active-active modes.

- Kubernetes networks. A cluster requires a minimum of two classless inter-domain routing (CIDR) ranges, one for the Kubernetes Services network, and another for the Kubernetes Pods network.

  NKE supports two container network interface (CNI) providers for the Kubernetes networks: Flannel and Calico.

  - *Flannel*. NKE uses the VXLAN mode. Changing the mode to host-gw is possible, but unsupported.
  - *Calico*. NKE uses the Direct mode. Changing the mode to IP in IP or VXLAN is possible, but unsupported.

Pro tip

You can leave the service CIDR and pod CIDR ranges as default, but the ranges must not overlap with each other or with an existing network in your data center if a pod in the cluster will require access to that external network.

Networking

A production cluster with active-active control plane mode requires an external load balancer.

Storage

When deploying a Kubernetes cluster, the Nutanix container storage interface (CSI) driver is also deployed along with it.

A default StorageClass is created as well during the deployment, which uses Nutanix Volumes. This is required by the included add-ons such as Prometheus for monitoring, and EFK (Elasticsearch, Fluent Bit, and Kibana) logging stack, to store metrics and logs. After deployment, more storage classes can be added using the same CSI driver (see examples).

Apart from Nutanix Volumes, you can also create a StorageClass for file storage using Nutanix Files. Depending on what storage backend is configured in a StorageClass, different access modes are supported when creating a PersistentVolumeClaim.

Storage backend	ReadWriteOnce RWO	ReadOnlyMany ROX	ReadWriteMany RWX	ReadWriteOncePod RWOP
Volumes	✓	✓	-	-
Files	✓	✓	✓	-

Access modes supported by CSI driver and storage backend.

Security

Access and Authentication

There are two components to keep in mind when it comes to access and authentication: NKE in PC, and an NKE-enabled Kubernetes cluster.

- NKE uses Prism Central authentication and RBAC. Nutanix requires configuring NKE users to a directory service in Prism Central like Microsoft Active Directory. Users can access the NKE console and perform certain tasks based on the assigned role.

  A member of the *User Admin* role in PC has full access to NKE and its functionalities.

  A member of the *Prism Central Admin* role or *Viewer* role can only download kubeconfig.

- An NKE-enabled Kubernetes cluster out-of-the-box uses Prism Central for authentication and maps the PC role *User Admin* with the Kubernetes role *cluster-admin*.

  From an authentication perspective, the Kubernetes cluster sends authentication requests to NKE which uses PC directory services. This means that you can authenticate your users against Active Directory out-of-the-box.

  From the RBAC standpoint, the PC role *User Admin* maps with the Kubernetes super-admin role named *cluster-admin*. This means that a user member of the *User Admin* role in PC is a super-user in all Kubernetes clusters managed by the NKE instance. On the other hand, the PC roles *Prism Central Admin* and *Viewer* do not have a mapping with a Kubernetes role. This means that a user member of any of these two roles can download the kubeconfig from NKE, but not perform any action at the Kubernetes level. A super-admin user will have to create the correct role mapping inside Kubernetes.

Blog

Providing RBAC for your Karbon Kubernetes Clusters

Note that the kubeconfig generated by NKE is valid for 24-hours, after which the user will have to request a new kubeconfig file. This can be done using the NKE GUI, CLI, API, or the kubectl plug-in available here (recommended).

Nodes

The SSH access to the Kubernetes nodes is locked down using an ephemeral certificate - available in the NKE console, which expires after 24-hours. Installing software or changing settings in the node OS is unsupported, changes are not persistent during upgrades or when scaling out a node pool. The only reason for accessing the nodes via SSH is for troubleshooting at the discretion of Nutanix support.

CIS Benchmark for Kubernetes

Nutanix has evaluated NKE-enabled Kubernetes cluster against the CIS Kubernetes Benchmark-1.6. You can verify compliance through Kube Bench, an automated open-source tool available on GitHub. The report is available here.

Add-ons

NKE add-ons are open source software extensions that provide additional features to your deployment. The add-ons are automatically installed when you deploy a Kubernetes cluster.

Nutanix Kubernetes Engine includes the following add-ons:

- A logging add-on powered by Elasticsearch, Fluent Bit, and Kibana (EFK)
- A monitoring add-on powered by Prometheus

These add-ons are for cluster internal use only. Their configuration is not designed for supporting the data generated by the applications running on the Kubernetes cluster. For collecting logs and metrics for the containerized applications, deploy dedicated instances of EFK and Prometheus, or re-use existing ones available in your environment.

Logging

The logging stack aggregates all the operating system and infrastructure logs from the Kubernetes nodes. The Kibana dashboard is accessible via the NKE console.

Since NKE 2.4, the logging stack can be disabled (more details) and just use Fluent Bit for log forwarding against an external existing logging stack (more details)

Monitoring

The Kubernetes clusters have the Prometheus operator installed and one instance of it deployed for collecting infrastructure metrics. Additional Prometheus instances can be deployed using the operator, for example, for application monitoring (blog).

Since NKE 2.4, SMTP-based alert forwarding to an e-mail address can be enabled (more details).

Lifecycle management

There are two different types of NKE upgrades:

- NKE version upgrades using the Life Cycle Management feature.
- Kubernetes cluster upgrades for node OS image, and Kubernetes version.

NKE upgrade via LCM

To check the current version of Karbon or to upgrade to later versions, perform the inventory check in Prism Central using LCM. LCM upgrades the following NKE components:

- NKE core (karbon-core container)
- NKE UI (karbon-ui container)

NKE Upgrades

Be aware when upgrading to a latest version of NKE, that all the Kubernetes clusters must be running or upgraded first to a supported version by the target NKE. Check the Nutanix portal for updated supported versions.

Kubernetes cluster upgrades

There are two aspects when it comes to upgrading a Kubernetes cluster:

- Node operating system upgrades
- Kubernetes + add-ons version upgrade

Kubernetes Cluster Upgrades

Be aware that node OS or Kubernetes version upgrades can be disruptive depending on your Kubernetes cluster type, development vs. production.

Node OS upgrade

When a node OS image upgrade is available, NKE displays an option to download the new image in the OS Images tab. NKE also displays an Upgrade Available icon next to the cluster in the Clusters view.

Kubernetes + add-ons version upgrade

Clusters that have a Kubernetes version eligible for an upgrade display the Upgrade Available icon in the table. As a part of the upgrade process, it will upgrade the Kubernetes version as well as any upgrade available for the installed add-ons.

NKE CLI and API

NKE CLI

The NKE CLI, karbonctl, gives users the ability to execute lifecycle management tasks for NKE and Kubernetes clusters. Certain advanced tasks can be done using karbonctl only.

To use karbonctl you have to SSH into a Prism Central instance. The path for the binary is /home/nutanix/karbon/karbonctl

Some common tasks you can run with karbonctl are:

- Configuring airgap deployment
- Configuring GPU support
- Rotating certificates
- Enabling/disabling alert forwarding
- Configuring a private registry
- Renewing the kubeconfig file

```
nutanix@NTNX-10-42-234-39-A-PCVM:~$ /home/nutanix/karbon/karbonctl
Karbonctl is a command line utility to manage your k8s clusters

Usage:
  karbonctl [command]

Available Commands:
  airgap      Used for Karbon Airgap configuration
  cluster     Used for k8s cluster specific operations
  help        Help about any command
  k8s         Used for getting the list of available k8s packages from the Nutanix Portal
  login       Generate a karbonctl configuration to allow passwordless authentication to Karbon
  os-image    Used for OS image management
  registry    Used for private registry operations
  version     Output of the karbonctl version information

Flags:
      --config string        Karbonctl configuration file path (default "/home/nutanix/.karbon/config/karbonctl.yaml")
  -h, --help                 help for karbonctl
      --output string        Supported output formats: ['default', 'json'] (default "default")
      --pc-ip string         Prism Central IP (default "127.0.0.1")
      --pc-password string   Password of the user in Prism Central
      --pc-port int          Prism port on the Prism Central VM (default 9440)
      --pc-username string   Username of the user in Prism Central
  -t, --toggle               Help message for toggle

Use "karbonctl [command] --help" for more information about a command.
```

NKE API

The NKE API lets users programmatically run management task for NKE and Kubernetes clusters. The API documentation is available at https://www.nutanix.dev/reference/karbon.