

# Ambiguous Terms

## A

<p><b>Authentication</b></p>	<p><b>Definition:</b> The process of verifying who the user is before granting access. It checks identity.</p> <p><b>Example:</b> “When you log in with your username and password, Prism Central authenticates your identity.”</p> <p><b>Tone:</b> Secure, procedural, identity-focused.</p> <p><b>Use when:</b> Describing login, credentials, or identity verification.</p>	<p><b>Conflated Terms:</b></p> <p>Authentication vs <a href="#">Authorization</a></p>
<p><b>Authorization</b></p>	<p><b>Definition:</b> The process of <b>deciding what the authenticated user can do</b> once verified. It checks permissions.</p> <p><b>Example:</b> “After authentication, authorization (<i>Scenario when a user needs permission from the admin to access a cluster</i>) determines whether the user can create VMs or only view them.”</p> <p><b>Tone:</b> Rule-based, permission-focused.</p> <p><b>Use when:</b> Describing <b>roles, policies, or access control</b>.</p>	<p><b>Conflated Terms:</b></p> <p>Authorization vs <a href="#">Authentication</a></p>
<p><b>Alert</b></p>	<p><b>Definition:</b> A notification or warning that draws attention but doesn’t always stop progress. An alert is something that needs attention.</p> <p><b>Example:</b> “Alert: Storage nearing capacity.”</p> <p><b>Use when:</b></p> <ul style="list-style-type: none"> <li>• The system detects a <b>condition that needs attention</b>, but the task can continue</li> <li>• The issue is <b>informational, warning-level, or preventive</b></li> <li>• You want to <b>notify or warn</b> the user before it becomes a failure</li> </ul> <p>Typical cases include thresholds, capacity nearing limits, expiring certificates, performance degradation, etc.</p>	<p><b>Conflated Terms:</b></p> <p>Alert vs <a href="#">Error</a></p>

# C

<p><b>Create</b></p>	<p><b>Definition:</b> Instantiate a new object/record and its settings. It may exist before resources are allocated.</p> <p><b>Example:</b> “Create a protection policy.”</p> <p><b>Tone:</b> Intent-setting, action-oriented</p> <p><b>Use when:</b></p> <ul style="list-style-type: none"> <li>• Introducing a <b>new object, record, or entity</b> for the first time</li> <li>• The user is <b>defining intent or structure</b> before resources are allocated</li> <li>• The action represents the <b>start of a lifecycle</b>, not execution or activation</li> </ul>	<p><b>Conflated Terms:</b></p> <p>Create vs <a href="#">Provision</a> vs <a href="#">Configure/Configured</a> vs <a href="#">Deploy</a></p>
<p><b>Configure/ Configured</b></p>	<p><b>Definition:</b> Set or change an object’s parameters; Configured typically means <b>settings are applied</b>. It is 1 task in a bigger activity.</p> <p><b>Example:</b> “Configure the snapshot schedule to proceed”.</p> <p><b>Tone:</b> Instructional; providing the user parameters to carry out tasks going forward.</p> <p><b>Use when:</b></p> <ul style="list-style-type: none"> <li>• Changing settings on something that already exists</li> <li>• Describing a <b>task-level step</b>, not an end-to-end process</li> </ul>	<p><b>Conflated Terms:</b></p> <p>Configure/Configured vs <a href="#">Create</a> vs <a href="#">Provision</a> vs <a href="#">Deploy</a></p>
<p><b>Control Plane</b></p>	<p><b>Definition:</b> The “brain” that manages and decides how traffic or resources should flow (routing tables, policies, metadata). In simpler words, control plane decides.</p> <p><b>Example:</b> “Prism Central acts as the control plane to manage multiple clusters.”</p>	<p><b>Conflated Terms:</b></p> <p>Control plane vs <a href="#">Management plane</a> vs <a href="#">Data plane</a></p>
<p><b>Control</b></p>	<p><b>Definition:</b> It is a mechanism for managing or enforcing with the aim of achieving a specific outcome. So control is a mechanism. It can involve rules, policies, guidelines, checks, prechecks, and other such entities that can be applied in specific sequences to corresponding outcomes.</p> <p><b>Example:</b> “A control checks if all critical VMs follow the backup policy.”</p> <p><b>Use when:</b> Referring to <b>monitoring, enforcement, or verification mechanisms.</b></p>	<p><b>Conflated Terms:</b></p> <p>Control vs <a href="#">Policy</a> vs <a href="#">Rule</a></p>

<p><b>Created</b></p>	<p><b>Definition:</b> The object or entity is brought into existence; it now exists in the system, but may not yet be ready for use.</p> <p><b>Example:</b> “VM created successfully.” The VM exists but may still need settings (such as network or storage) before it can run.</p> <p><b>Use when:</b> Referring to the initial creation or provisioning of a resource.</p>	<p><b>Conflated Terms:</b></p> <p>Created vs <a href="#">Configured</a></p>
<p><b>Configured</b></p>	<p><b>Definition:</b> The object has been set up or customized with the required parameters to make it functional.</p> <p><b>Example:</b> “Network configured with IP ranges and VLANs. It is now ready for workloads.”</p> <p><b>Use when:</b> Referring to settings, tuning, or readiness after creation.</p>	<p><b>Conflated Terms:</b></p> <p>Configured vs <a href="#">Create</a></p>
<p><b>Container (Kubernetes)</b></p>	<p><b>Definition:</b> Containers are a lightweight way to isolate applications using the operating system. A container packages an application together with its required files, libraries, and dependencies, so it runs the same way across different environments. <b>Kubernetes manages these containers</b>, handling deployment, scaling, and reliability across multiple machines. Use “<b>Kubernetes container</b>” for runtime contexts</p> <p><b>Example:</b> “Each microservice in your app runs inside its own Kubernetes container, managed automatically by Kubernetes.”</p> <p><b>Use when:</b></p> <ul style="list-style-type: none"> <li>Referring to <b>application execution</b></li> <li>Talking about <b>microservices, pods, or app runtime behavior</b></li> </ul> <p><i>Note: Always qualify the word <b>container</b> in UI and docs. Never use “<b>container</b>” alone when both concepts exist in the same product area.</i></p>	<p><b>Conflated Terms:</b></p> <p>Container (Kubernetes) vs <a href="#">Storage Container</a></p>

## D

<p><b>Define</b></p>	<p><b>Definition:</b> To set or specify something clearly.</p> <p><b>Example:</b> “Define the configuration settings.” You are explicitly stating what the settings are.</p> <p><b>Tone:</b> Instructional, like establishing a rule.</p> <p><b>Use when:</b></p> <ul style="list-style-type: none"> <li>You are explicitly specifying values, rules, or settings</li> </ul>	<p><b>Conflated Terms:</b></p> <p>Define vs <a href="#">Determine</a></p>
----------------------	--	---

	<ul style="list-style-type: none"> <li>• The system needs clear inputs from the user</li> <li>• The action is about setting up or declaring a configuration, not deciding outcomes</li> </ul>	
<b>Determine</b>	<p><b>Definition:</b> To decide or control the outcome of something.</p> <p><b>Example:</b> “Protection policy Recovery Point Objective (RPO) determines your maximum data loss during failover.”</p> <p><b>Use when:</b></p> <ul style="list-style-type: none"> <li>• Describing <b>how one setting affects or controls an outcome</b></li> <li>• Explaining <b>cause-and-effect relationships</b></li> <li>• The result is <b>derived or influenced</b>, not directly set by the user</li> </ul>	<p><b>Conflated Terms:</b></p> <p>Determine vs <a href="#">Define</a></p>
<b>Data Plane</b>	<p><b>Definition:</b> The “conduit or tubes” that actually move the traffic or data as instructed by the control plane. In simpler words, data plane does</p> <p><b>Example:</b> “The data plane transfers VM traffic across nodes.” For example, for the Network Controller in Flow Virtual Networking, the Network Controller is the control plane, the PC is the Management plane, and the OVS bridges or cluster Virtual Switches are the data plane.</p>	<p><b>Conflated Terms:</b></p> <p>Data plane vs <a href="#">Control Plane</a> vs <a href="#">Management Plane</a></p>
<b>Dashboard</b>	<p><b>Definition:</b> A dashboard is the primary control surface for a product. All controls, all settings, widgets, and an overview of the stats within that product. It is the control room with all the information for a specific product, so users can monitor and act quickly.</p> <p><b>Example:</b> “The Prism Central dashboard shows cluster health, alerts, capacity, and key actions.”</p> <p><b>Use when:</b></p> <ul style="list-style-type: none"> <li>• This is the main landing area for a product or module</li> <li>• Users need both visibility and actions</li> <li>• Multiple widgets or sections coexist</li> </ul>	<p><b>Conflated Terms:</b></p> <p>Dashboard vs <a href="#">Overview</a> vs <a href="#">Details</a></p>
<b>Deploy</b>	<p><b>Definition:</b> To set up and install a feature, service, or component so it becomes available for use.</p> <p><b>Example:</b> “Deploy Prism Central on the management cluster.”</p> <p><b>Use when:</b> Something is being installed, instantiated, or made ready in an environment.</p>	<p><b>Conflated Terms:</b></p> <p>Deploy vs <a href="#">Create</a> vs <a href="#">Provision</a> vs <a href="#">Configure/Configured</a></p>

<p><b>Disable</b></p>	<p><b>Definition:</b> Disable means to intentionally turn off a feature, capability, or control so it cannot operate until it's enabled again. It does not delete anything, and does not power down a system.</p> <p><b>Example:</b> “Disable alerts for this cluster during maintenance.”</p> <p><b>Tone:</b> Neutral, controlled, administrative. Implies a reversible, intentional restriction, often for safety or maintenance.</p> <p><b>Use when:</b></p> <ul style="list-style-type: none"> <li>• Stopping a feature, rule, or capability The system or resource should still exist and remain powered on</li> <li>• The action is reversible</li> </ul>	<p><b>Conflated Terms:</b></p> <p>Disable vs <a href="#">Turn on</a> vs <a href="#">Turn off</a> vs <a href="#">Enable</a></p>
<p><b>Delete</b></p>	<p><b>Definition:</b> Permanently erase an object and all associated data. It cannot be recovered (unless backed up).</p> <p><b>Example:</b> “Delete the snapshot to free up storage space.</p> <p><b>Use when:</b> The object and its data are <b>completely removed</b> from the system.</p>	<p><b>Conflated Terms:</b></p> <p>Delete vs <a href="#">Remove</a> vs <a href="#">Unregister</a></p>
<p><b>Details</b></p>	<p><b>Definition:</b> Details provide the complete, in-depth information about a specific entity. This view focuses on attributes, configuration, state, and metadata, rather than summaries or KPIs.</p> <p><b>Example:</b> “Open VM details to view CPU, memory, disks, network settings, and events.”</p> <p><b>Use when:</b></p> <ul style="list-style-type: none"> <li>• Users need to inspect or troubleshoot</li> <li>• Showing full configuration or properties</li> <li>• Supporting advanced or admin-level tasks</li> </ul>	<p><b>Conflated Terms:</b></p> <p>Details vs <a href="#">Dashboard</a> vs <a href="#">Overview</a></p>

## E

<p><b>Enable</b></p>	<p><b>Definition:</b> To turn on or activate a deployed or existing feature so it starts functioning. In simpler words, enable is to activate and make operational.</p> <p><b>Example:</b> “Enable microsegmentation after deploying the security service.” Use when the</p>	<p><b>Conflated Terms:</b></p> <p>Enable vs <a href="#">Turn on</a> vs <a href="#">Turn off</a> vs</p>
----------------------	--	--

	<p>feature is already deployed and needs to be switched on or configured for use.</p> <p><b>Use when:</b></p> <ul style="list-style-type: none"> <li>• The feature is present but inactive</li> <li>• You want to make it operational without deploying anything new</li> </ul>	<a href="#">Disable</a>
<b>Ephemeral Storage</b>	<p><b>Definition:</b> <b>Ephemeral storage</b> is <b>temporary storage</b> that exists only while a workload (VM, container, or pod) is running. When the workload stops, restarts, or is deleted, <b>the data is lost.</b></p> <p><b>Example:</b> “Use ephemeral storage for caching during jobs.”</p> <p><b>Use when:</b></p> <ul style="list-style-type: none"> <li>• Data is short-lived</li> <li>• When data can be recreated</li> <li>• When performance matters more than durability</li> </ul>	<p><b>Conflated Terms:</b></p> <p>Ephemeral Storage vs <a href="#">Persistent Storage</a></p>
<b>Error</b>	<p><b>Definition:</b> A problem that <i>blocks an action</i> and needs fixing before proceeding. An error is something that went wrong.</p> <p><b>Example:</b> “Error: Invalid password.”</p> <p><b>Use when:</b></p> <ul style="list-style-type: none"> <li>• An action cannot be completed as requested</li> <li>• The system hits a blocking condition or failure</li> <li>• User intervention is required to proceed</li> </ul> <p>Typical cases include failed operations, invalid inputs, unreachable services, and permission issues.</p>	<p><b>Conflated Terms:</b></p> <p>Error vs <a href="#">Alert</a></p>

## H

<b>Host</b>	<p><b>Definition:</b> A host is a hypervisor-enabled node that runs one or more VMs. It provides compute, memory, and network connectivity to workloads. <b>host is the hypervisor running on that server.</b></p> <p><b>Example:</b> “Each host runs multiple VMs using the AHV hypervisor.”</p> <p><b>Use when:</b> Referring to a virtualization context. E.g., the node’s role in running VMs or containers under a hypervisor.</p>	<p><b>Conflated terms:</b></p> <p>Host vs <a href="#">Node</a></p>
-------------	---	--

<b>Help</b>	<p><b>Definition:</b> Provides <b>instructional or problem-solving information</b> to guide the user when they're stuck, uncertain, or need step-by-step direction.</p> <p><b>Example:</b> "Need help setting up a Protection Policy? See the setup guide."</p> <p><b>Use when:</b></p> <ul style="list-style-type: none"> <li>• Offering <b>clarification, how-to steps, or links to documentation.</b></li> <li>• Helps users complete or understand a task.</li> <li>• <b>Help</b> appears in tooltips, side panels, or "Learn more" links for setup or error resolution.</li> </ul>	<p><b>Conflated terms:</b></p> <p>Help vs <a href="#">Tip</a></p>
-------------	---	---

## I

<b>Image</b>	<p><b>Definition:</b> It is a <b>disk file</b> (such as a virtual hard disk or ISO) that contains an operating system or application. It's used to create or boot virtual machines. An image <b>does not define VM sizing or network settings.</b></p> <p><b>Example:</b> "Upload a Nutanix AOS image to the Image Service, then use it to deploy Prism Central or create a new VM."</p> <p><b>Use when:</b></p> <ul style="list-style-type: none"> <li>• Referring to the <b>OS or software media</b></li> <li>• Talking about <b>what the VM boots from</b></li> </ul>	<p><b>Conflated Terms:</b></p> <p>Image vs <a href="#">Template</a> vs <a href="#">Profile</a></p>
--------------	--	--

## M

<b>Management Plane</b>	<p><b>Definition:</b> The management plane is where administrators control, configure, and monitor the system. It's responsible for managing infrastructure, not running workloads. In simpler words, <b>management plane</b> is where one manages and controls the system.</p> <p><b>Example:</b> "Prism Central operates in the management plane, handling configuration, upgrades, and policies for clusters."</p>	<p><b>Conflated Terms:</b></p> <p>Management plane vs <a href="#">Control plane</a> vs <a href="#">Data plane</a></p>
-------------------------	---	---

## N

<p><b>Node</b></p>	<p><b>Definition:</b> A node is a physical server in a Nutanix cluster that contributes compute, storage, and networking resources. Multiple nodes work together to form a cluster. In simpler words, a <b>node is the physical server</b>.</p> <p><b>Example:</b> “Add two more nodes to increase the cluster’s capacity and resilience.”</p> <p><b>Use when:</b> Use Node when referring to hardware components, the actual servers running AHV (or other hypervisors) that participate in the cluster.</p>	<p><b>Conflated terms:</b></p> <p>Node vs <a href="#">Host</a></p>
<p><b>Network</b></p>	<p><b>Definition:</b> A broader collection of connected devices and systems that can communicate with each other. In Nutanix, a network often refers to the logical or physical setup that VMs or clusters use to connect. In simple words, a <b>network</b> is the <i>overall communication space</i>.</p> <p><b>Example:</b> “Create a network for all VMs that need access to the corporate datacenter.”</p> <p><b>Use when:</b></p> <ul style="list-style-type: none"> <li>• Referring to the <b>entire connectivity environment</b></li> <li>• Talking at a <b>high level</b> about communication or isolation</li> </ul>	<p><b>Conflated terms:</b></p> <p>Network vs <a href="#">Subnet</a></p>

## O

<p><b>Overlay</b></p>	<p><b>Definition:</b> Overlay is a logical (virtual) network built on top of the underlay.</p> <p>It enables features such as network isolation, segmentation, and multi-tenancy without requiring changes to the physical hardware.</p> <p><b>Example:</b> “VMs communicate securely across datacenters using an Overlay network.”</p> <p><b>Use when:</b> Talking about virtual networking, segmentation, or software-defined networks.</p>	<p><b>Conflated Terms:</b></p> <p>Overlay vs <a href="#">Underlay</a></p>
<p><b>Overview</b></p>	<p><b>Definition:</b> An overview is a high-level snapshot that highlights the most important information at a glance. It usually appears within a dashboard and focuses on KPIs or critical status indicators.</p> <p><b>Example:</b> “The Overview widget shows cluster health, alerts, and resource usage.”</p>	<p><b>Conflated Terms:</b></p> <p>Overview vs <a href="#">Dashboard</a> vs <a href="#">Details</a></p>

	<p><b>Use when:</b></p> <ul style="list-style-type: none"> <li>• Users need to be given a quick understanding without deep detail</li> <li>• Showing KPIs, health, or status</li> <li>• As an entry point before deeper exploration</li> </ul>	
--	--	--

## P

<b>Provision</b>	<p><b>Definition:</b> Allocate and configure the resources needed for use (E.g., compute, storage, network, identity). Makes the object runnable. In other words, it is a set of configuration tasks.</p> <p><b>Example:</b> "Provision a VM with 4 vCPUs, 8 GiB RAM, and VLAN 120." OR "Provision a Database using a particular software."</p> <p><b>Use when:</b> You imply a larger entity, like a cluster, which includes setting up the resources.</p>	<p><b>Conflated Terms:</b></p> <p>Provision vs <a href="#">Create</a> vs <a href="#">Configure/Configured</a> vs <a href="#">Deploy</a></p>
<b>Power Off</b>	<p><b>Definition:</b> An immediate stop that cuts power to the VM, like pulling a plug. It doesn't give the OS time to close processes. It can cause data loss or corruption.</p> <p><b>Example:</b> "Power off this VM if it's unresponsive and shutdown is not working."</p> <p><b>Use when:</b> Power Off is triggered as an emergency or admin override action.</p>	<p><b>Conflated Terms:</b></p> <p>Power Off vs <a href="#">Shutdown</a></p>
<b>Persistent Storage</b>	<p><b>Definition:</b> Persistent storage is durable storage that keeps data even after a workload stops, restarts, or moves. It's used for data that must be saved long-term. In simpler words, with the help of persistent storage, data stays even when the workload stops</p> <p><b>Example:</b> "Attach persistent storage to ensure DB data remains intact and is retained."</p>	<p><b>Conflated Terms:</b></p> <p>Persistent Storage vs <a href="#">Ephemeral Storage</a></p>
<b>Policy</b>	<p><b>Definition:</b> A <b>set of rules or guidelines</b> defined by the user or admin, describing <i>what should happen</i>.</p> <p><b>Example:</b> "Create a data protection policy to take VM backups every 6 hours."</p> <p><b>Use when:</b> Referring to a <b>user-defined objective or set of conditions</b> that guide automation or compliance. It is <i>the rulebook</i>.</p>	<p><b>Conflated Terms:</b></p> <p>Policy vs <a href="#">Rule</a> vs <a href="#">Control</a></p>

<p><b>Profile</b></p>	<p><b>Definition:</b> It represents a specific configuration. So when a master profile changes, all child entities inherited from it update automatically. It's dynamic and live.</p> <p>Remember: if you change settings in the parent profile, child profiles get changed as well (dynamic).</p> <p>It's a reusable set of configuration settings that's present and helps standardize how resources are configured (for example, VM sizing, network rules, or environment settings). It focuses on consistency and governance.</p> <p><b>Example:</b> "Apply the VM profile to ensure all database VMs use the same CPU and memory settings."</p> <p><b>Use when:</b></p> <ul style="list-style-type: none"> <li>• Enforcing standard configurations</li> <li>• Applying policies or presets across multiple resources</li> </ul> <p><b>Note:</b> <i>Often mistaken for a template. A profile <b>defines settings</b>.</i></p>	<p><b>Conflated Terms:</b></p> <p>Profile vs <a href="#">Image</a> vs <a href="#">Template</a></p>
-----------------------	---	--

## R

<p><b>Resize</b></p>	<p><b>Definition:</b> <b>Resize</b> means <b>changing the size of an existing resource</b> (CPU, memory, or disk) without creating new ones, which can mean up <b>or</b> down.</p> <p><b>Example:</b> "Resize the disk to 1 TB to handle more data."</p> <p><b>Use when:</b></p> <ul style="list-style-type: none"> <li>• Adjusting the <b>capacity or performance</b> of an existing resource</li> <li>• The <b>resource remains the same</b>, only its size changes</li> <li>• Scaling <b>up or down within the same object</b> (not adding more objects)</li> </ul>	<p><b>Conflated Terms:</b></p> <p>Resize vs <a href="#">Scale out</a></p>
<p><b>Remove</b></p>	<p><b>Definition:</b> Detach or take something out of its current context <b>without destroying it</b>.</p> <p><b>Example:</b> "Remove a node from the cluster; it stays intact, and can be added again later."</p> <p><b>Use when:</b> The object <b>still exists</b>, but no longer belongs to the group or configuration.</p>	<p><b>Conflated Terms:</b></p> <p>Remove Vs <a href="#">Delete</a> vs <a href="#">Unregister</a></p>
<p><b>Review</b></p>	<p><b>Definition:</b> A review is a confirmation step that asks the</p>	<p><b>Conflated Terms:</b></p>

	<p>user to pause and verify their inputs before completing an action. It emphasizes intent and correctness, not just information.</p> <p><b>Example:</b> “Review these settings before creating the VM.”</p> <p><b>Tone:</b> Cautious, intentional</p> <p><b>Use when:</b></p> <ul style="list-style-type: none"> <li>• The action is <b>high-impact or hard to undo</b>, and errors would be costly</li> <li>• You want to encourage <b>deliberate checking</b></li> </ul> <p><b>Note:</b> <i>Preview is sometimes used instead of Review, but for consistency, use Review. Also, "review" is a more widely understood and commonly used term.</i></p>	<p>Review vs <a href="#">Summary</a></p>
<b>Rule</b>	<p><b>Definition:</b> A <b>specific condition or statement</b> inside a policy that dictates how it should behave.</p> <p><b>Example:</b> “Rule: Take snapshots every 6 hours and retain for 7 days.”</p> <p><b>Use when:</b> Describing <b>individual directives within a policy</b>. It is the specific instruction inside the rulebook.</p>	<p><b>Conflated Terms:</b></p> <p>Rule vs <a href="#">Policy</a> vs <a href="#">Control</a></p>
<b>Reset</b>	<p><b>Definition:</b> Reset means to return something to a default or known state. This may clear settings, data, or user-defined values.</p> <p><b>Example:</b> “Reset network settings to default.”</p> <p><b>Use when:</b></p> <ul style="list-style-type: none"> <li>• Fixing misconfiguration</li> <li>• Undoing user changes</li> <li>• Recovering from errors</li> </ul>	<p><b>Conflated Terms:</b></p> <p>Reset vs <a href="#">Restart</a></p>
<b>Restart</b>	<p><b>Definition:</b> Restart means to stop and start something again (a VM, service, node, or app) without changing its configuration.</p> <p><b>Example:</b> “Restart the VM to apply the configuration changes.”</p> <p><b>Use when:</b></p> <ul style="list-style-type: none"> <li>• You want changes to take effect</li> <li>• The system is running, but needs a fresh start</li> <li>• No data or settings should be lost</li> </ul>	<p><b>Conflated Terms:</b></p> <p>Restart vs <a href="#">Reset</a></p>

# S

<p><b>Shutdown</b></p>	<p><b>Definition:</b> A controlled process where the guest OS safely closes applications and services before turning off. Data is saved, and integrity is maintained.  <b>Example:</b> “Shutdown this VM to stop it gracefully and preserve application state.”</p> <p><b>Tone:</b> Safe, recommended, system-friendly.</p> <p><b>Use when:</b> Shutdown is an <b>operational</b> and safe management action.</p>	<p><b>Conflated Terms:</b></p> <p>Shutdown vs <a href="#">Power off</a></p>
<p><b>Scale out</b></p>	<p><b>Definition:</b> Add more VMs to expand capacity horizontally by increasing the number of resources.</p> <p><b>Example:</b> “Scale out the cluster by adding two more nodes.”</p> <p><b>Use when:</b></p> <ul style="list-style-type: none"> <li>• You need more capacity or throughput</li> <li>• The system supports horizontal expansion</li> <li>• Adding more VMs, pods, or nodes, not changing existing ones</li> </ul> <p><b>More info:</b></p> <p>a) Scale-out (hyphenated) Use the <b>hyphenated version</b> when the term is used as an <b>adjective or noun</b></p> <ul style="list-style-type: none"> <li>• “Scale-out architecture”</li> <li>• “This feature supports scale-out growth.”</li> </ul> <p>b) Scale out (two words) Use the unhyphenated, two-word version when it functions as a verb</p> <ul style="list-style-type: none"> <li>• “You can scale out the cluster.”</li> <li>• “The system will scale out automatically.”</li> </ul> <p><b>Note:</b> <i>There is no term as Scale Down, it’s always resize. Scale out is always used for VMs.</i></p>	<p><b>Conflated Terms:</b></p> <p>Scale out vs <a href="#">Resize</a></p>
<p><b>State</b></p>	<p><b>Definition:</b> Describes the <b>current condition or phase</b> an object is in. It’s often tied to its <b>lifecycle</b>.</p> <p><b>Example:</b> “VM state: Powered On / Powered Off / Suspended.”</p> <p><b>Use when:</b> Referring to <b>operational or lifecycle stages</b> (running, paused, deleted, active).</p>	<p><b>Conflated Terms:</b></p> <p>State vs <a href="#">Status</a></p>

<b>Status</b>	<p><b>Definition:</b> Indicates the <b>health, result, or quality</b> of that object or process.</p> <p><b>Example:</b> “Cluster status: Healthy / Warning / Critical.”</p> <p><b>Use when:</b> Referring to <b>health or outcome</b> (healthy, failed, pending).</p>	<p><b>Conflated Terms:</b></p> <p>Status vs <a href="#">State</a></p>
<b>Summary</b>	<p><b>Definition:</b> A summary is a concise recap of the most important information or changes. It focuses on what has been selected or configured, without asking the user to think deeply or double-check.</p> <p><b>Example:</b> “VM Summary: 4 vCPUs, 16 GB memory, 2 disks.”</p> <p><b>Use when:</b></p> <ul style="list-style-type: none"> <li>• Information can be compressed and scanned quickly</li> <li>• The action is low to medium risk</li> <li>• You want to inform, not slow the user down</li> </ul> <p><b>Note:</b> <i>Summary is used mostly as headers or tab names</i></p>	<p><b>Conflated Terms:</b></p> <p>Summary Vs <a href="#">Review</a></p>
<b>Subnet</b>	<p><b>Definition:</b> A smaller, defined segment of a network with its own IP address range. Subnets are used to organize, isolate, and manage traffic. In simple words, a subnet is <i>a slice of a network</i>.</p> <p><b>Example:</b> “Create a subnet to separate database and application VM traffic.”</p> <p><b>Use when:</b></p> <ul style="list-style-type: none"> <li>• You need <b>traffic separation, IP management, or security boundaries</b></li> <li>• Designing network layouts within the same network</li> </ul>	<p><b>Conflated Terms:</b></p> <p>Subnet vs <a href="#">Network</a></p>
<b>Storage Container</b>	<p><b>Definition:</b> A <b>storage container</b> is a logical storage space within a Nutanix cluster that stores <b>VM disks, snapshots, and volume data</b>. It organizes how data is stored and managed on the cluster.</p> <p><b>Example:</b> “Create a storage container for production VM disks.”</p> <p><b>Use when:</b></p> <ul style="list-style-type: none"> <li>• Referring to where data is stored</li> <li>• Talking about storage organization, capacity, or replication</li> </ul>	<p><b>Conflated Terms:</b></p> <p>Storage Container vs <a href="#">Container (Kubernetes)</a></p>

# T

<p><b>Turn on</b></p>	<p><b>Definition:</b> Start or activate something so it begins working or becomes available.</p> <p><b>Example:</b> “Turn on replication to start protecting your workloads.”</p> <p><b>Use when:</b></p> <ul style="list-style-type: none"> <li>• A feature, setting, or service is <b>currently inactive</b> and needs to be <b>enabled</b>.</li> <li>• The action is <b>reversible</b> (users can turn it off later).</li> </ul>	<p><b>Conflated Terms:</b></p> <p>Turn on vs <a href="#">Turn off</a> vs <a href="#">Enable</a> vs <a href="#">Disable</a></p>
<p><b>Turn Off</b></p>	<p><b>Definition:</b> Stop or deactivate something so it no longer runs or consumes resources.</p> <p><b>Example:</b> “Turn off alerts to stop receiving notifications.”</p> <p><b>Use when:</b> A feature, service, or toggle is <b>active</b> and should be <b>disabled</b></p>	<p><b>Conflated Terms:</b></p> <p>Turn off vs <a href="#">Turn on</a> vs <a href="#">Enable</a> vs <a href="#">Disable</a></p>
<p><b>Tip</b></p>	<p><b>Definition:</b> Offers <b>useful, optional advice or optimization</b>. Not required to complete the task, but helps users do it better or faster.</p> <p><b>Example:</b> “Tip: Group VMs by category to simplify reporting.”</p> <p><b>Use when:</b></p> <ul style="list-style-type: none"> <li>• Providing <b>best practices or shortcuts</b>, not troubleshooting.</li> <li>• Tip enhances user success, not fixes confusion.</li> <li>• Tip appears in info banners or carousels offering admin-friendly best practices (e.g. “<i>Tip: Schedule snapshots during off-peak hours.</i>”).</li> </ul>	<p><b>Conflated Terms:</b></p> <p>Tip vs <a href="#">Help</a></p>
<p><b>Template</b></p>	<p><b>Definition:</b> A template is a blueprint that, for example, combines an image with predefined configuration (CPU, memory, disks, network) and is used to create an entity. Subsequent changes to the template do not affect the already created entity. <b>Templates are static and initial.</b></p> <p>Remember: if you change anything in the parent template, other templates do not change (static)</p> <p><b>Example:</b> A template is like a 'replica' of the VM itself. It</p>	<p><b>Conflated Terms:</b></p> <p>Template vs <a href="#">Image</a> vs <a href="#">Profile</a></p>

	<p>contains everything required to create another VM. A template includes image and configuration, plus it's ready to deploy.</p> <p>“Create a VM from the web-server template.”</p> <p><b>Use when:</b></p> <ul style="list-style-type: none"> <li>• Referring to <b>reusable configuration profiles</b>.</li> <li>• Creating <b>new VMs repeatedly</b></li> <li>• The aim is <b>speed + consistency</b> in deployment</li> </ul>	
--	--	--

## U

<p><b>Underlay</b></p>	<p><b>Definition:</b> The physical network that carries switches, routers, and cabling that provide IP connectivity. Underlay is a physical foundation</p> <p><b>Example:</b> “The underlay provides physical connectivity between all cluster nodes.”</p> <p><b>Use when:</b> Talking about <b>physical networking, like IP addressing, routing, MTU, cabling, and switches.</b></p>	<p><b>Conflated Terms:</b></p> <p>Underlay vs <a href="#">Overlay</a></p>
<p><b>Unregister</b></p>	<p><b>Definition:</b> To <b>remove an object from a list, registry, or association</b> so it's no longer tracked or referenced. In simple words, unregistering is simply disconnecting or removing a relationship.</p> <p><b>Example:</b> “Unregister a cluster from Prism Central. It will no longer be managed, but continues to run independently.”</p> <p><b>Use when:</b></p> <ul style="list-style-type: none"> <li>• The object remains <b>active elsewhere</b>, just not managed by this system.</li> <li>• The action feels <b>reversible or low-risk</b></li> </ul> <p><b>Note:</b> Always use <i>unregister</i> and not <b><u>deregister</u></b>, as it's more common in the industry, and also within Nutanix documentation. <i>Unregister also works better in correlation with register.</i></p>	<p><b>Conflated Terms:</b></p> <p>Unregister vs <a href="#">Remove</a> vs <a href="#">Delete</a></p>
<p><b>Update</b></p>	<p><b>Definition:</b> Apply <b>minor changes or patches</b> to improve stability, security, or performance, without changing the product version. Update is a small fix.</p> <p><b>Example:</b> “Update Prism Central to the latest patch for bug fixes.”</p>	<p><b>Conflated Terms:</b></p> <p>Update vs <a href="#">Upgrade</a></p>

	<b>Use when:</b> Describing <b>small, incremental improvements</b> within the same release.	
<b>Upgrade</b>	<p><b>Definition:</b> Move to a <b>newer version</b> of a product or component that adds features or major improvements. Upgrade is a major version change.</p> <p><b>Example:</b> “Upgrade Prism Central from version 2023.3 to 2024.1.”</p> <p><b>Use when:</b> Describing <b>major version changes</b> that can alter functionality or UI.</p>	<p><b>Conflated Terms:</b></p> <p>Upgrade vs <a href="#">Update</a></p>

## V

<b>View</b>	<p><b>Definition:</b> A saved way of looking at data, often combining filters, sorts, or layouts. In simple words. A <b>view</b> is about how data is presented. <b>Views</b> help <b>determine what data one wants to see</b>, such as filtering or grouping of information.</p> <p><b>Example:</b> Create a view to track only production VMs with high CPU usage.</p>	<p><b>Conflated Terms:</b></p> <p>View vs <a href="#">Widget</a></p>
-------------	--	--

## W

<b>Widget</b>	<p><b>Definition:</b> A rectangular object that provides dynamically updated information at a glance, often includes a data visualization, and may offer an option to interact with the data, for example, by adjusting the data time period or moving the widget around a dashboard. <b>Widgets</b> are the <b>visual blocks</b> (charts, tables, or tiles) that <b>show data</b> on a dashboard.</p> <p><b>Example:</b> Add a widget showing cluster storage usage over time.</p>	<p><b>Conflated Terms:</b></p> <p>Widget vs <a href="#">View</a></p>
---------------	---	--

# Nutanix Internal Terminologies

For teams working on products at Nutanix

## A: Layers at Nutanix

Nutanix separates infrastructure management from how users consume resources. This ensures scalability, governance, and clear ownership.

Admins manage the underlying infrastructure, while users deploy and operate workloads within controlled environments. Together, this enables centralized control with flexible, self-service consumption.

### A1. Infrastructure Layer

**Definition:** It is the foundational resource that powers the platform, such as compute, storage, and networking. This layer is responsible for provisioning and maintaining day 2 operations, such as adding more nodes to clusters, updating networks, etc.

**Usage:** Used when referring to provisioning, managing, or maintaining the underlying infrastructure.

**Example:** An admin provisions a cluster with compute, storage, and networking resources.

### A2. User / Consumption Layer

**Definition:** The interface where users interact with the platform to deploy, manage, and consume resources. This layer is based on the scope that the user has to operate in, which is defined by the project or IAM. It is always the IAM admin (who is today a super admin) who configures access. and day-to-day operations.

**Usage:** Used when describing how developers, DBAs (database admins), or operators, app owners access and use platform resources.

**Example:** A developer deploys a database or VM within the Payments project to run their application. Users can operate either **manually** (e.g., creating a VM within a project scope) or through **automation and templates**, depending on how workflows are set up for consistency and scale.

## B: Governance at Nutanix

**Nutanix is moving away from manual governance.** In other words, Nutanix is providing new tools and constructs to enable large enterprises to govern enterprises at scale and move away from error-prone manual governance.

Manual governance is a model where administrators manually manage policies, access, and resources across the entire platform.

As environments grow (multiple clusters, teams, and workloads), this approach becomes slow, inconsistent, and difficult to scale.

### B1. Global Governance

**Definition:** Establishes and enforces consistent, centralized policies and controls at the platform level, applying universally across infrastructure, projects, and users. Note: all global governance is envisioned

to happen at the NC level.

**Usage:** It is used when referring to platform-wide rules or standards set by administrators.

**Example:** A platform administrator implements a mandatory, company-wide policy requiring all database resources to utilize encryption.

## B2. Defined User Boundaries

**Definition:** Role-based permissions control what users can access or modify.

**Usage:** It is used when defining user roles, access levels, and operational limits.

**Example:** A developer can manage VMs in their project, but cannot modify infrastructure.

**Definition:** The **App Domain** is where **user applications and workloads** run databases, VMs, web servers, etc. It is isolated from management operations for security and performance.

**Usage:** Use when talking about the environment where business workloads live.

## C. Domain

The word “**Domain**” is used inconsistently:

- A **Prism Central** is referred to as “Domain” in Nutanix Central
- Sometimes, Nutanix Central (NC) and Nutanix Cloud Manager (NCM) is termed as a domain
- Sometimes management vs app domains

**Note:** There could be regular PCs in the absence of Nutanix Infrastructure Manager (NIM), and if a customer is using NIM, there will be management and app domains configured. But any Prism Central which is not doubling up as a management or an app domain is simply termed as a Domain on Nutanix Central.

So Nutanix Central will have:

- Management Domains (the environment, in other words, the set of clusters)
- Application Domains
- Domains (in the absence of NIM, domains are not classified as management or application domains)

### C1. Management Domain

**Definition:** A **management domain** is a logical grouping of management-only clusters managed by a single management Prism Central, where **management components/services** such as Prism Central, Nutanix Central (NC), and Nutanix Cloud Manager (NCM), etc. run. It also hosts Application Domain PCs.

The management domain itself does not perform management; instead, it contains these management services, which in turn manage other app domains and clusters.

**Example:** A large enterprise will have a central hub where it deploys management-domain Prism Central to host all Nutanix services and keep compute, storage, and network resources separate from its user workloads.

**Usage:** Referring to a PC that hosts Nutanix services and application domain PCs.

**Ambiguity:** Often confused with Prism Central domain or application domain.

The management Prism Central **runs within the management domain cluster**, not in the domain itself. Application domains host **workloads/applications**, whereas management domain hosts **management services**, and the **application domain Prism Centrals**

**One-liner:** The management domain runs management services.

## C2. Application Domain

**Definition:** An application domain is a logical boundary. It comprises of an Application Domain Prism Central and the workload clusters registered to it. It is here that user applications and workloads (VMs, databases, containers) run. The Application domain Prism Central is deployed onto clusters within the Management Domain.

**Example:** “Run the Payments and Analytics applications in separate App Domains to isolate workloads and ensure performance.”

**Usage:** Use while talking about the environment where business workloads live.

**Ambiguity:** Sometimes used interchangeably with “cluster,” but application domains can **span clusters**.

**One-liner:** Application domain is where end-user workloads (*i.e., your apps and services*) run. These are the apps that admins deploy (as VMs) using the management services.

## C3. Fault Domain

**Definition:** A fault domain is a smaller failure zone inside a cluster (node, rack, or block). Nutanix spreads replicas across fault domains to maintain resilience.

**Example:** “Nutanix automatically distributes VM replicas across fault domains to ensure data safety.”

**Usage:** Use while referring to intra-cluster resilience and replica placement.

**Ambiguity:** Frequently confused with “Availability Zone” since both relate to failure boundaries.

**One-liner:** Fault Domain is a smaller fault boundary.

## C4. Protection Domain:

**Definition:** A defined group of entities (VMs and volume groups) that are backed up locally on a cluster and can optionally be replicated to one or more remote sites on a schedule for disaster recovery

**Note:** *We refer to NCM/NKP as domains today. It would thus be a good idea to add a prefix for clarity.*

**Usage:** Use when referring to grouping workloads for backup, replication, and disaster recovery policies.

**Example:** “Add critical VMs to a protection domain to schedule regular backups and replicate them to a remote site for disaster recovery.”

### Ambiguity:

- Often confused with Fault Domain and Availability Zone
  1. A fault domain is like a boundary of infra components, like nodes, clusters, that share a common risk of failure. It can be at a rack level, a node level, an availability zone level, or even a data center level.
  2. Availability Zone provides failure isolation across sites
  3. A protection domain is data protection and DR grouping, not a failure boundary
- The term “domain” suggests infrastructure grouping, but here it refers to data protection scope

## D. Project

**Definition:** A project is the fundamental unit of consumption in which **resources, entities, and workloads** are defined and managed for a set of users that are added to the project. Projects are application-specific, meaning each project type is created to support a particular product, use case, and set of entities. It provides a **logical boundary that separates infrastructure from workload consumption**.

**Usage:** Use **Project** when referring to the **environment where teams deploy and manage workloads**, such as VMs, databases, or Kubernetes applications.

Projects define:

- resource scope
- user access
- quotas and policies for workloads

**Example:** In a payments app, the resources inside the project would include:

- Payment service VMs
- Payment databases
- Networking and policies

Developers and operators working on the payment application are given access to this project.

### Types of Projects

#### D1. Prism Central Project

**Definition:** Created within Prism Central and scoped to Prism Central specific entities (clusters, VMs, etc.). Previously referred to as “local projects.”

**Usage:** Use while referring to **projects that manage on-prem or Prism Central–scoped resources** such as clusters, VMs, and networking within a single PC.

**Example:** “Create a Prism Central project to manage VMs and networks for the Payments application within PC-A.”

**One-liner:** Prism Central Project basically manages workloads within a single Prism Central.

**Note:** *Since platform/domain/Prism Central /local projects all essentially refer to the same thing today, it only makes sense to simply term it as Prism Central Project. Since Prism Central is the only widely used term, it would reduce confusion*

## D2. Global Project (Multi-domain)

**Definition:** it is authored in Nutanix Central and spans multiple PC domains.

Each project:

- Serves as a specific product or workload.
- Owns its entities, configurations, and scope
- Is managed independently by default.

**Usage:** Use while referring to projects that need to **operate across multiple Prism Central domains**, while still **maintaining independent ownership** and **configuration** per workload or product.

**Example:** “The Payments global project spans PC-A and PC-B, allowing teams to manage workloads across both environments.”

**Note:** *The definition for Global Projects may evolve with time. Currently, Global Project is also referred to as Cloud Projects, but that may be renamed going forward. Once that happens, we will update it.*

## D3. NCM Project

**Definition:** Created for Nutanix Cloud Manager use cases. Includes cloud accounts, remote PCs, and other NCM-managed entities.

**Usage:** Use while referring to **projects that span cloud and hybrid environments**, including cloud accounts, remote Prism Centrals, and NCM-managed resources.

**Example:** “Create an NCM project to manage workloads across AWS and on-prem clusters from a single interface.”

**One-liner:** NCM Projects manages workloads across cloud and multi-environment setups.

## D4. Scoped Consumption (within projects and IAM)

**Definition:** A user's scope of operation can either be defined within IAM or within projects today.

- **Projects**

**In projects, consumption of platform resources** is restricted and managed within specific, predefined boundaries (e.g., individual projects) rather than being available for use across the entire platform.

**Usage:** Use when describing **how user access and actions are restricted to specific resources or boundaries**.

**Example:** “The Analytics team can deploy and manage VMs only within the Analytics project, based on their assigned scope.”

- **IAM**

**In IAM, access is based on two things: their roles and scope–**

**Roles include:**

- a) System roles like Project Admin or Project Viewer with predetermined permissions (based on the operations present in the role)
- b) Custom roles (with granular operations added by the IAM admin.)

**Scope includes:**

- a) All VMs in a specific Prism Central (for example, PC-A)
- b) All VMs within a specific category in a Prism Central
- c) All VMs in a category across multiple Prism Centrals (with Global IAM)
- d) All VMs across multiple Prism Centrals (with Global IAM)

**Usage:** Used when describing how users access and deploy resources within controlled boundaries.

**Example:** The Analytics team can deploy VMs only within the Analytics project.

**One-line contrast:** Projects are fixed scope by resources; IAM is flexible scope by rules.

**Key difference (simplified)**

Aspect	Scoped Consumption under Projects	Scoped Consumption under IAM
Scope type	Fixed	Flexible
Based on	Assigned resources	Rules (categories, PCs, etc.)
Reach	One project	Can span multiple projects or Prism Centrals
Control style	Boundary	Filter

## E: Workspace

### Why Workspaces Exist

Large customers often have:

- Many clusters
- Multiple Prism Centrals
- Hundreds of projects across products

Managing permissions, policies, categories, or templates on a project-by-project basis does not scale.

A workspace solves this by allowing users to:

- Set permissions once, at the workspace level
- Share policy templates and categories centrally
- Perform common platform actions across all included projects

These settings are then propagated automatically to every project in that workspace.

### Workspace Across Nutanix Products

The same word “**Workspace**” already exists in **three different contexts** within Nutanix. These represent **different constructs**.

#### E1. MyNutanix Workspace

- **Definition:** A **scope switcher** used to separate environments, files, and user actions. Similar to Figma or Slack workspaces
- **Usage:** Used to switch between organizational contexts within MyNutanix.
- **Example:** Switching between **Personal Workspace** and **Company Workspace**. Similar to Slack or Figma workspaces.

#### E2. NKP (Kubernetes Workspace)

- **Definition:** A grouping construct used to organize **Kubernetes clusters and applications**.
- **Usage:** NKP workspaces can be used both as tenant or team boundaries, and enable fleet management of all Kubernetes clusters within that Workspace.
- **Example:** A workspace containing multiple **Kubernetes clusters for a development environment**.
- **Note:** Conceptually closer to a **Project** than to the new NC Workspace.

## F. Zones

A Zone is a logical grouping of a management domain and all its Application Domains; this can appear in Nutanix Central as a new entity.

This term also has potential for confusion. Today, we have “Availability Zone,” which is another term for a Prism Central.

**One-liner:** A Zone is a collection of management plus application domains

#### F1. Infrastructure Zone (earlier Infrastructure Domain)

**Definition:** An infrastructure zone is a logical boundary that groups one or more co-located Nutanix clusters so they can operate with a local control plane while still being governed by a central management plane. It enables site-local runtime operations with centralized visibility, policy, and lifecycle management.

- Can be accessed/run independently even if Prism Central is down or disconnected from it
- Runs its independent set of control and control plane services
- Typically mapped to physical failure domains (DCs, server rooms, racks, etc.)

**Example:** A retailer can create one infrastructure zone for each regional data center or store group, so local VM, networking, DR, and security operations continue even if connectivity to Prism Central is slow or temporarily unavailable.

**Use when:** Use it when you need central control with local independence—especially for distributed or edge environments.

**One Liner:** A collection of clusters that is an independent control plane within a Prism Central.

## F2. Availability Zone

**Definition:** An availability zone is a large failure-isolation zone, like a datacenter, site, or cloud availability zone. Each domain runs independently, so one site's failure doesn't affect another.

**Example:** "Spread clusters across availability zones for site-level fault tolerance."

**Use when:** Referring to multi-site or multi-region redundancy.

**Ambiguity:** Commonly mixed up with "Fault Domain." In Nutanix, an availability zone is datacenter-level isolation. AWS uses the same term ("Availability Zone") but with a region-based connotation.

**One-liner:** Availability zone is an independent zone for high availability. In other words, it's a collection of infrastructure zones (or clusters) that form an independent failure boundary at a site level.