

Book of Cloud Management - Intelligent Operations

[PDF generated December 27 2023. 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.]

In the following sections we'll cover some key features that are part of Nutanix Intelligent Operations.

NOTE: Not all features are currently covered. In this section, we currently cover:

- [Anomaly Detection](#)
- [Capacity Planning](#)
- [Low-code/no-code Automation with X-Play](#)

Stay tuned for updates for the rest of the features in Intelligent Operations:

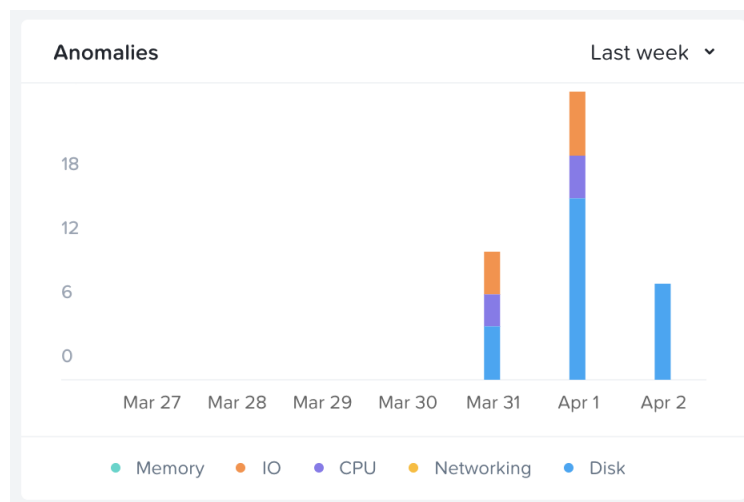
- Reporting
- Non-Nutanix ESXi Monitoring (i.e. monitoring 3-tier ESXi)
- SQL Monitoring
- Application Discovery
- Self-Tuning with Machine Learning (X-Pilot)

Anomaly Detection

In the world of IT operations there is a lot of noise. Traditionally systems would generate a great deal of alerts, events and notifications, often leading to the operator either a) not seeing critical alerts since they are lost in the noise or b) disregarding the alerts/events.

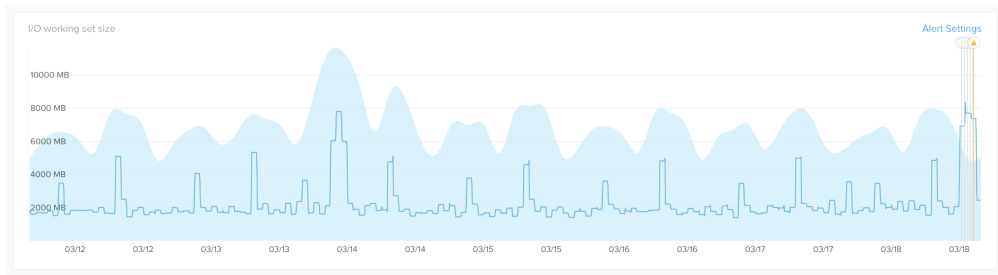
With the Anomaly Detection feature of Intelligent Operations, the system will monitor seasonal trends for time-series data (e.g. CPU usage, memory usage, latency, etc.) and establish a "band" of expected values. Only values that hit outside the "band" will trigger an event / alert. You can see the anomaly events / alerts from any entity or events page.

The following chart shows a lot of I/O and disk usage anomalies as we were performing some large batch loads on these systems:



Anomaly Chart

The following image shows the time-series values for a sample metric and the established "band":

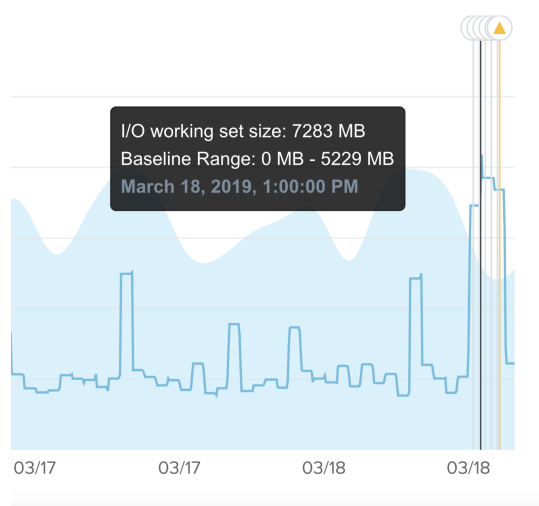


Anomaly Band

This reduces unnecessary alerts as we don't want alerts for a "normal" state. For example, a database system will normally run at >95% memory utilization due to caching, etc. In the event this drops to say 10% that would be an anomaly as something may be wrong (e.g. database service down).

Another example would be how some batched workloads run on the weekend. For example, I/O bandwidth may be low during the work week, however on the weekends when some batch processes run (e.g. backups, reports, etc.) there may be a large spike in I/O. The system would detect the seasonality of this and bump up the band during the weekend.

Here you can see an anomaly event has occurred as the values are outside the expected band:



Anomaly Event

Another topic of interest for anomalies is seasonality. For example, during the holiday period retailers will see higher demand than other times of the year, or during the end of month close.

Anomaly detection accounts for this seasonality and leverages the following periods to compare between micro (daily) and macro (quarterly) trends:

- Daily
- Weekly
- Monthly

You can also set your own custom alerts (also called user-defined alerts), that enables you to alert on behavioral anomalies or static thresholds:

Create Alert Policy ? X

Entity Type: **VM**

Entity: **One VM**

Entity ID: **prd-pc-01**

Metric: **I/O working set size**

Impact Type: **Performance**

Policy Name: **prd-pc-01 - VM I/O working se**

Description: *Optional

Auto resolve alerts

Enable Policy

I/O working set size 6 Anomalies Last week

Behavioral Anomaly

Every time there is an anomaly, alert **Warning**

Ignore all anomalies between and

Static Threshold

Alert Critical if or

Alert Warning if or

Trigger alert if conditions persist for: **10 Minutes**

Cancel **Save**

Anomaly Custom Event

Anomaly Detection Algorithm

Nutanix leverages a method for determining the bands called 'Generalized Extreme Studentized Deviate Test'. A simple way to think about this is similar to a confidence interval where the values are between the lower and upper limits established by the algorithm.

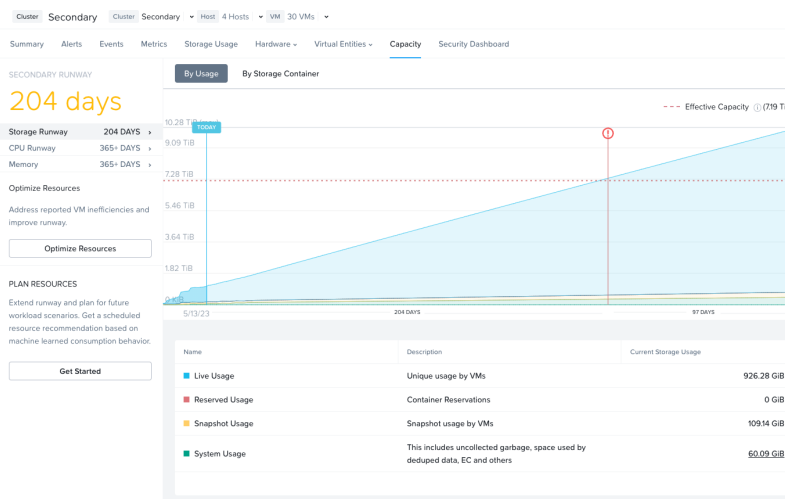
The algorithm requires 3 x the granularity (e.g. daily, weekly, monthly, etc.) to calculate the seasonality and expected bands. For example, the following amounts of data would be required to adapt to each seasonality:

- Daily: 3 days
- Weekly: 3 weeks (21 days)
- Monthly: 3 months (90 days)

Twitter has a good resource on how they leverage this which goes into more detail on the logic: [LINK](#)

Capacity Planning

To get detailed capacity planning details you can click on a specific cluster under the 'cluster runway' section in Prism Central to get more details:



Capacity Planning

This view provides detailed information on cluster runway and identifies the most constrained resource (limiting resource). You can also get detailed information on what the top consumers are as well as some potential options to clean up additional capacity or ideal node types for cluster expansion.

Optimize Resources ? x

Recommendation [Get Report](#)

Consider optimizing over provisioned VMs, deleting inactive VMs and adding resource to constrained VMs.

<p>24</p> <p>Overprovisioned</p> <p>Potential Reclaim: 162.49 GiB Memory, 88 vCPU</p>	<p>2</p> <p>Inactive</p> <p>Potential Reclaim: 10 GiB Memory, 3.46 GiB Storage, 6 vCPU</p>
<p>0</p> <p>Constrained</p>	<p>0</p> <p>Bully</p>
<p>View All Inefficient VMs</p>	

Capacity Planning - Recommendations

X-Play

When we think about our daily activities the more we can automate the better. We are constantly doing this in our daily lives with our routines and technology enables us to do the same in other areas. Nutanix X-Play (pronounced CrossPlay) allows us to automate a common set of activities. However, before diving into the product, let's first cover what we're trying to do.

Event driven automation works in the following manner:

event(s) → logic → action(s)

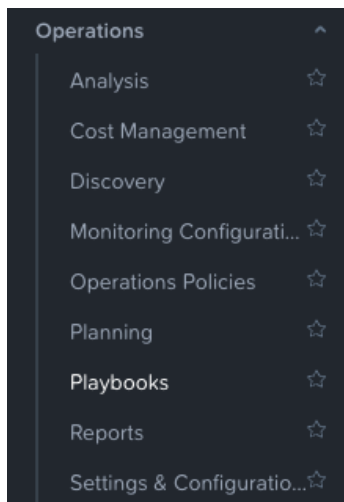
In this scenario there's some sort of event (or cascading events) that occur which triggers a series or set of actions. A great example of this is [IFTTT](#) which takes an event, applies some logic (hence the 'if this then that' acronym), then performs some action.

For example, take turning off the lights at home when we leave. If we can program the event (e.g. leaving home / device not present) to trigger the system to turn off all the lights automatically, that makes our lives much simpler.

If we compare this to our IT operations activities we see a similar pattern. An event occurs (e.g. a VM needs more disk space) and then we perform a series of actions (e.g. create a ticket, add storage, close ticket etc.). These repetitive activities are a perfect example of where automation can add value and enable us to focus on more beneficial activities.

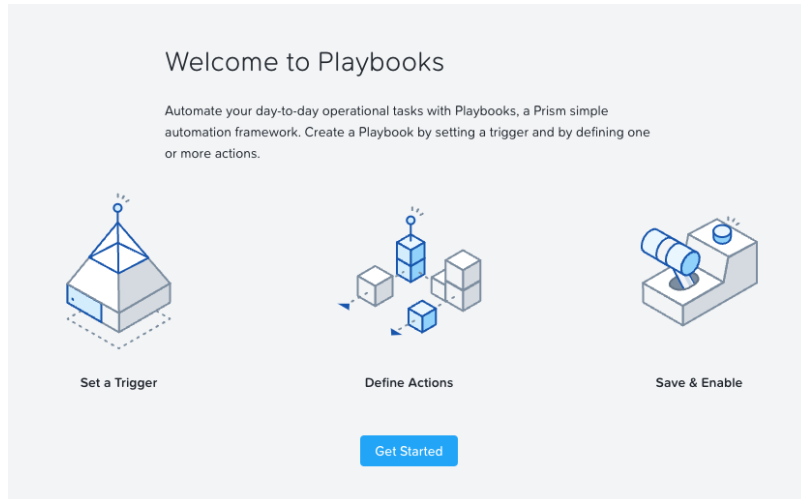
With X-Play we can take a series of events / alerts and allow the system to intercept those and perform a series of actions.

To get started navigate to the 'Plays' section under 'Operations' in the Infrastructure menu:



X-Play - Navigation

This will launch the main X-Play page:



X-Play - Playbooks Overview

Click on 'Get Started' to view the current plays and/or create a new one:

Playbooks | List | Plays | Action Gallery 14 Total Playbooks

Create Playbook | Import | Actions

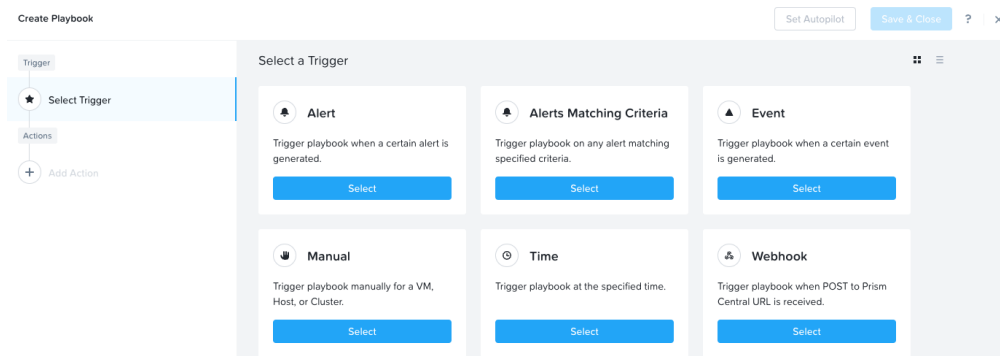
Type text to filter by Modify Filters

Viewing all 14 Playbooks Export | 1 - 14 of 14 | 20 per page

<input type="checkbox"/>	Name	Trigger	Status	Autopilot Status	Owner	Run on behalf of	Last Updated Time
<input type="checkbox"/>	VM Created Slack Alert - Imported (17-05-23, 18:25:13Z)	Event	Enabled	Disabled	admin	admin	05/18/23, 03:29:49 PM
<input type="checkbox"/>	VM Deleted Slack Alert - Imported (17-05-23, 18:23:10Z)	Event	Enabled	Disabled	admin	admin	05/17/23, 11:33:34 AM
<input type="checkbox"/>	xplay-report	Manual	Enabled	Disabled	admin	Triggering User	05/16/23, 03:54:01 PM
<input type="checkbox"/>	VM Create Send E-mail	Event	Enabled	Disabled	admin	admin	05/16/23, 03:38:11 PM
<input type="checkbox"/>	manualRP-with-TTL2	Manual	Enabled	Disabled	adminuser08@ntnvlab.local	Triggering User	05/10/23, 12:53:20 PM
<input type="checkbox"/>	manualRP-with-TTL	Manual	Disabled	Disabled	adminuser08@ntnvlab.local	Triggering User	05/10/23, 12:49:17 PM
<input type="checkbox"/>	low cpu	Alert	Validation Errors	Disabled	adminuser04@ntnvlab.local	adminuser04@ntnvlab.local	05/09/23, 04:14:24 PM
<input type="checkbox"/>	test	Manual	Disabled	Disabled	adminuser07@ntnvlab.local	Triggering User	05/09/23, 01:56:26 PM
<input type="checkbox"/>	Add New Xplay VM to Category	Event	Enabled	Disabled	admin	admin	05/04/23, 02:29:43 PM
<input type="checkbox"/>	Report a cluster resource shortage <small>System</small>	Alerts Matching Criteria	-	Disabled	admin	-	04/25/23, 07:37:02 AM
<input type="checkbox"/>	Reduce memory for overprovisioned VMs <small>System</small>	Alert	-	Disabled	admin	-	04/25/23, 07:37:02 AM
<input type="checkbox"/>	Reduce CPU for overprovisioned VMs <small>System</small>	Alert	-	Disabled	admin	-	04/25/23, 07:37:02 AM
<input type="checkbox"/>	Increase memory for constrained VMs <small>System</small>	Alert	-	Disabled	admin	-	04/25/23, 07:37:02 AM
<input type="checkbox"/>	Increase CPU for constrained VMs <small>System</small>	Alert	-	Disabled	admin	-	04/25/23, 07:37:02 AM

X-Play - Playbooks

From here you can create a new playbook by first defining the trigger:



X-Play - Trigger

The following shows an example trigger which is based upon a user-defined alert policy:

Alert

[Change Trigger](#)

Alert Policy

AppType:Earth_Stack - VM Memory Usage

Filter Criteria

The selected policy has the following criteria defined. You could refine the criteria further to target the right alerts and right entities.

Severity

Critical

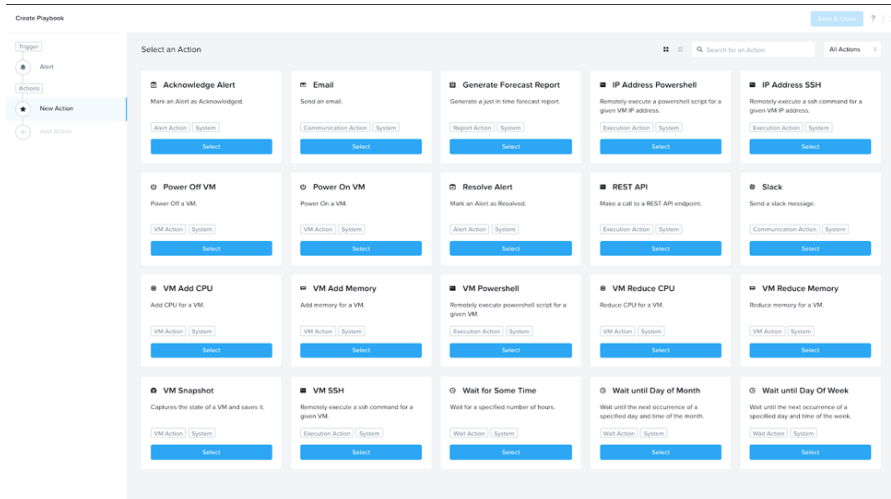
Target VM

All Applicable VMs

Specify VMs

X-Play - Trigger - User Defined Alert

Once the trigger has been defined, you now specify a series of actions. The following shows some sample actions:



X-Play - Actions

You then input the details for the action, this shows a sample REST API call:

REST API

[Change Action](#)

Method: POST

URL: https://myurl.nutanix.com/myEndpoint

Username: foo

Password: ...

Request Body:

```
{
  "customTicket": {
    "entity": "[trigger(0).source_entity_info.name]",
    "eventTime": "[trigger(0).creation_time]",
    "eventType": "[trigger(0).alert_entity_info.name]"
  }
}
```

Request Headers:

Content-Type: application/json

Settings: On Action Failure: Stop

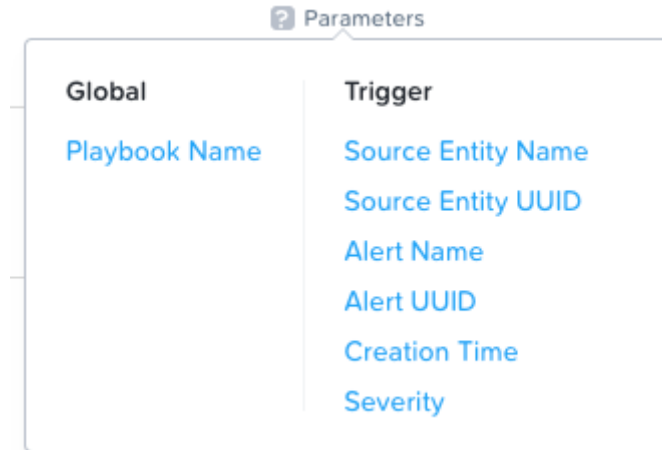
X-Play - Sample REST Action

REST API Actions and External Systems

X-Play provides a multitude of default actions like sending email, sending a slack message, as well as others like performing a REST API call.

This is critical when we think about interfacing with external systems like a CMDB or other ticketing / automation tools. By using a REST API action we can interface with those to create / resolve tickets, kick off other workflows, etc. This is an extremely powerful option as it enables all systems to be in sync.

For entity / event specific details you can use the 'parameters' variables which will give you details about the event, entity and others:



X-Play - Action Parameters

Once complete you can save you play and it will start to execute as defined.

The following shows a sample play whith multiple actions performed:

The screenshot shows the X-Play interface for a play named "Increase a constrained VM's memory". The left sidebar shows the play's name, description, last updated time (03/13/19, 7:09:08 PM), created time (01/02/19, 5:48:58 PM), and status (-). The main area shows the play configuration with a "Trigger" section set to "Alert" and an "Alert" section with details: Alert Policy (VM (vm_name) Memory Constrained), Source Entities (All Applicable VMs), and Severity (Info). The "Actions" section lists: VM Snapshot, VM Add Memory, Resolve Alert, and Email.

X-Play - Sample Playbook

The plays tab will show execution time of the play and status:

The screenshot shows the "Plays" tab in the X-Play interface. It displays a table with the following data:

Playbook	Trigger	Status	Occurrence Time	Last Updated Time
Earth_Stack: Memory Alert	Alert	Succeeded	03/14/20, 08:52:50 AM	03/14/20, 08:52:50 AM
Earth_Stack: Memory Alert	Alert	Succeeded	03/14/20, 08:52:50 AM	03/14/20, 08:52:50 AM

X-Play - Plays Executed

For some example playbooks that you can import into your own instance, check out the [Playbooks Library](#) on nutanix.dev!

Also, check out [Getting Started with Nutanix X-Play](#) for more examples and references.