

Turbonomic - Performance

Sample Demo Script for the live demo environment

©2023 Joël Hartmann/Niklaus Hirt / IBM

1. Introduction

This script is intended as a guide to demonstrate Turbonomic using the live demo environment. The script is presented in a few sections. You can utilize some or all sections depending upon your client's needs.

The script is intended to be used with the [Click Through PPT](#), but should work with the live Turbonomic demo environment that you can reserve via [TechZone](#).

In the demo script,

- “ **Action**” denotes a setup step for the presenter.
- “ **Narration**” denotes what the presenter will say.
- “ **Note**” denotes where the presenter may need to deviate from this demo script or add supplemental comments.

1.1 Key Terminology

You should be familiar with the following terminology when discussing Turbonomic:

- **Application Resource Management:** is a top-down, application-driven approach that continuously analyzes applications' resource needs and generates fully automatable actions to ensure applications always get what they need to perform. It runs 24/7/365 and scales with the largest, most complex environments.

To perform Application Resource Management, Turbonomic represents your environment holistically as a supply chain of resource buyers and sellers, all working together to meet application demand. By empowering buyers (VMs, instances, containers, and services) with a budget to seek the resources that applications need to perform, and sellers to price their available resources (CPU, memory, storage, network) based on utilization in real-time, Turbonomic keeps your environment within the desired state — operating conditions that achieve the following conflicting goals at the same time:

Assured application performance

Prevent bottlenecks, upsize containers/VMs, prioritize workload, and reduce storage latency.

Efficient use of resources

Consolidate workloads to reduce infrastructure usage to the minimum, downsize containers, prevent sprawl, and use the most economical cloud offerings.

- **Business Application:** A Business Application is a logical grouping of application entities and nodes that work together to compose a complete application as end users would view it. Turbonomic users can monitor overall performance, make resourcing decisions, and set policies in the context of their Business Applications.
- **Market:** The Turbonomic Market is an abstraction that represents the datacenter as buyers and sellers in a supply chain. Each entity (such as physical machines, virtual machines, storage device, volume, application component) in the environment is a buyer or seller. The Turbonomic Supply Chain is a graphical display of the buyer and seller relationships. Turbonomic uses Virtual Currency to give a budget to buyers and assign cost to resources. This virtual currency assigns value across all tiers of your environment, making it possible to compare the cost of application transactions with the cost of space on a disk or physical space in a datacenter.
- **Target:** A Target is a resource or workload management service in your virtual environment that you have connected to Turbonomic. For example, a public cloud account on AWS (Amazon Web Services) can be a target, as can an on-prem datacenter managed by VMware vCenter Server. For each target that you configure, Turbonomic communicates with the service via the management protocol that it exposes — a REST API, SMI-S, XML, or some other management transport. Turbonomic uses this communication to discover the managed entities, monitor resource utilization, and execute actions.
- **Commodity:** The basic building block of Turbonomic supply and demand. All the resources that Turbonomic monitors are commodities. For example, the CPU capacity or memory that a host can provide are commodities. Turbonomic can also represent clusters and segments as commodities. When the user interface shows commodities, it's showing the resources a service provides. When the interface shows commodities bought, it's showing what that service consumes.

- **Consumes:** The services and commodities a service has bought. A service consumes other commodities. For example, a VM consumes the commodities offered by a host, and an application consumes commodities from one or more VMs. In the user interface you can explore the services that provide the commodities the current service consumes.
- **Entity:** A buyer or seller in the market. For example, a VM or a datastore is an entity.
- **Environment:** The totality of data center, network, host, storage, VM, and application resources that you are monitoring.
- **Inventory:** The list of all entities in your environment.
- **Risk Index:** A measure of the risk to Quality of Service (QoS) that a consumer will experience. The higher the Risk Index on a provider, the more risk to QoS for any consumer of that provider's services. For example, a host provides resources to one or more VMs. The higher the Risk Index on the provider, the more likely that the VMs will experience QoS degradation. In most cases, for optimal operation the Risk Index on a provider should not go into double digits.

1.2 Demonstration flow

1. Scenario introduction
2. Understanding and optimising the infrastructure
 1. Introduce the demo context
 2. Login to Turbonomic
 3. Overview of the SupplyChain
 4. Integrating new Target
 5. Explanation Commodities and Optimization
 6. Zoom into Business Application
 7. Drill-down Actions
 8. Explain Plan Management
3. Summary

2. Deliver the demo

2.1 Introduce the demo context

Narration

Welcome to this demonstration of the Turbonomic platform. In this demo, I am going to show you how Turbonomic can help your operations team proactively optimize performance across mission-critical workloads, while improving cost.

Let's imagine we're at work, it's end of the year, and I get an email from Uwe, our Group CTO, saying that we have massive performance problems with our customer facing applications and that customers are unhappy.

Because of this we get way too many tickets and our support is overwhelmed with customer complaints.

So he suggests that we onboard our customer facing store to the Turbonomic solution that we just acquired.

Obviously, Black Friday is an absolute priority for our end-of-year sales strategy, accounting for a significant 40% of our total sales during the period spanning from Black Friday to the end of the year.

Note

During the demonstration, we will be using the sample application called RobotShop, which serves as a proxy for any type of app. The application is built on a microservices architecture, and the services are running on Kubernetes cluster. All actions performed on Turbonomic, like adding the RobotShop application, are purely for demonstration purposes and **must not be executed**.

Action

Use the [Click Through PPT](#) to get started with the demo

Narration

Now let's start the demo.

2.2 Login to Turbonomic (only on live Environment)

Narration

Let's take a closer look at Turbonomic.

To get started with the platform, I will log in and get started managing my environment.

Action

- Navigate your Web browser to the Turbonomic installation
- Provide the user name and password for your account

2.3 Overview of the SupplyChain

Narration

The supply chain represents the infrastructure of my complete IT landscape.

So I can see that I have 65 **BusinessApplications**, that are composed of **Services**, **ApplicationComponents** and **VirtualMachines** that are running in **VirtualDataCenters** or in the cloud in different **Zones** and **Regions**.

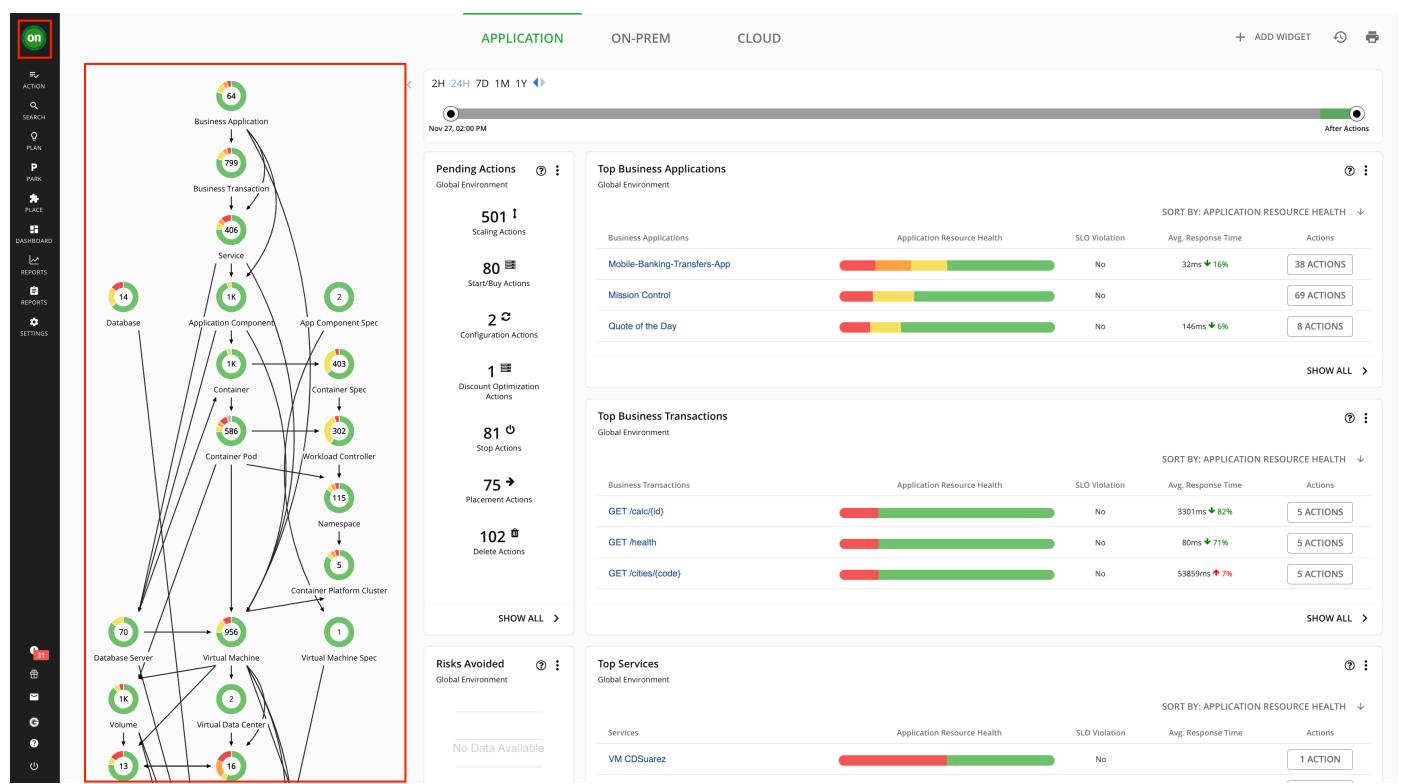
An **ApplicationComponent**, for example, could be a database running in a VM or it could very well also be, let's say, a web server running in a container. So the details are completely abstracted away.

The same goes for virtual machine as well: So we have 946 virtual machines and that can be a virtual machine running on a vSphere on-prem or this could be a virtual machine running on Azure or on AWS or in Google Cloud. It doesn't matter because those differences get abstracted away and normalized in order to get this unified holistic view.

So we get a complete overview of our infrastructure, be it on-prem or off-prem, in the private or public cloud.

Action

In the Demo UI, click **Turbonomic** on the top left



Narration

So now the first thing that I have to do is to onboard my RobotShop application that is not being managed by Turbonomic.

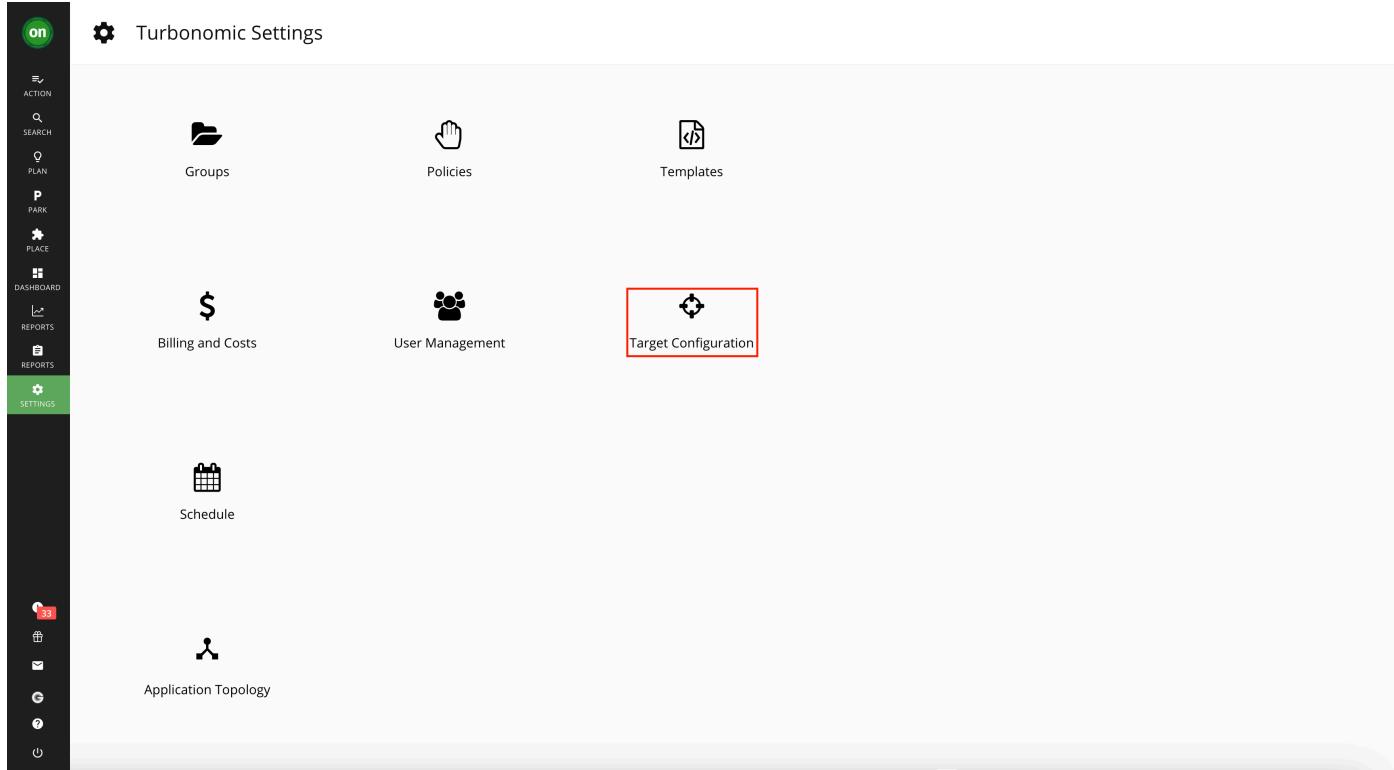
It's pretty simple to construct this supply chain. Now let's see how we add additional elements to Turbonomic.

2.4 Integrating new Target

Action

Click **Settings**

Click **Target Configuration**



Narration

So I'm going to my settings, target configuration.

Here I can see that I have different targets that have already been configured like vCenter connections, Kubernetes Clusters, Google Cloud, Azure and AWS Endpoints and Datadog or Instana Monitoring Tools for example. And I can monitor their status at any time.

So these Targets as we call them provide all your infrastructure components.

Now I have to create a new target to add in my Sales Application.

Let me just show you, how this is done.

So for our application, for example, I just have to add a target connection to the vSphere Server. For this I just have to give it the server URL, the user name and the password to connect to the Server.

Here we can use a user that already has read and write access.

When we start off for POVs or POCs for example, we might as well chose a read-only user so that we can just start reading and making recommendations but not change anything in your vSphere or back-end or in your public cloud accounts.

 **Action**

Click **New Target**

Click **Hypervisor**

Click **vCenter**

Add **Credentials**

Click **Add**

 **Narration**

Now our RobotShop application will be added as a target. It will take 10-15 minutes for its elements to be added to Turbonomic. Then we will be able to see the first recommendations.

2.5 Explanation Commodities and Optimization

Action

In the Demo UI, click **Turbonomic** on the top left to get back on main page

Narration

So a little bit later I get back and my shopping application has now been onboarded and I will see some recommended actions getting generated that will help make sure that my performance will be at the level we expect it to be.

So how does that work?

First let me show you quickly how those recommendations are being created.

If we go, for example, into **VirtualMachines** and just pick one **VirtualMachine** here.

You can see that we have changed perspective, specifically for the selected **VM**.

So I can see that my **VM** (here Turbo 897 test) is running in a **VirtualDataCenter** and on a specific host, I get the complete context here of my VM.

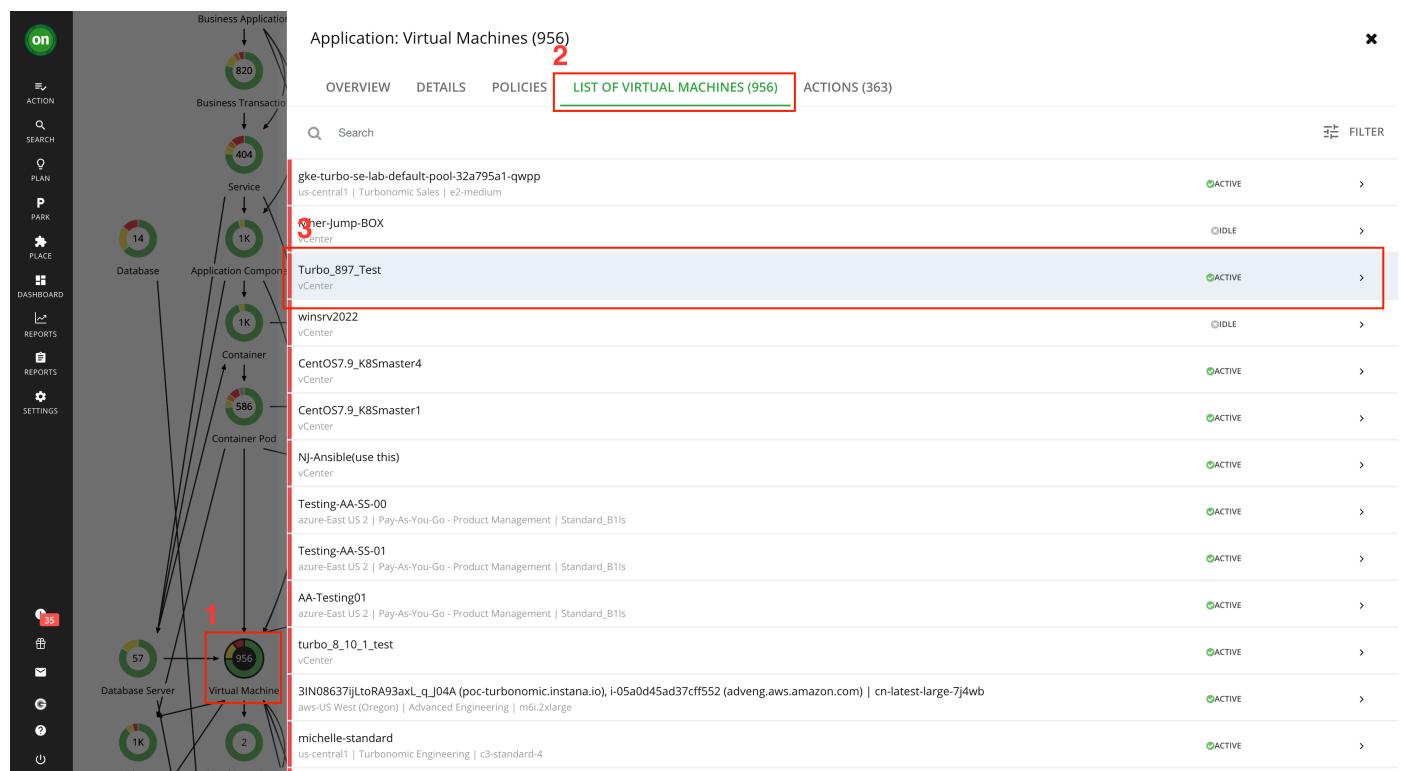
Let's take a closer look at how recommended actions are being created in Turbonomic.

Action

Click **Virtual Machine** on the supply chain

Click **List of Virtual Machines**

Click **Turbo_897_Test** to select this VM.



Narration

Let's take a look at **Capacity and Usage**.

Action
Click Capacity and Usage

The screenshot shows the Turbonomic interface for managing a virtual machine named "Turbo_897_Test". The main dashboard displays a resource dependency graph with nodes: Virtual Machine (1), Virtual Data Center (1), Host (1), Storage (1), and Network (1). The Storage node has a "Volume" dependency with a count of 4. The Host node has a "Disk Array" dependency with a count of 1. The Virtual Machine node has dependencies on the Virtual Data Center and Host. The Virtual Data Center node has dependencies on the Storage and Host nodes. The Storage node has a dependency on the Disk Array with a count of 1. The Network node has no dependencies.

Pending Actions:

- Resize up vMEM for Virtual Machine Turbo_897_Test from 4 GB to 6 GB (VMEM Congestion in Virtual Machine Turbo_897_Test)
- Resize down vCPU for Virtual Machine Turbo_897_Test from 2 to 1 vCPUs (CPU Ready risk reduction, Underutilized vCPU in Virtual Machine Turbo_897_Test)

Health:

- 1 VM with Critical Risks

Entity Information:

ID	75214064253312
Name	Turbo_897_Test
OS and Software	CentOS 7 (64-bit)
Type	On-Prem
Target Name	dedwdc06-vc.na.clo

Capacity and Usage:

Commodity	Capacity	Used	Utilization
Virtual Memory	4 GB	2.31 ...	57.71%
Virtual Storage	796.66 M	182....	22.85%
Virtual Storage	2.99 GB	663....	21.68%
Virtual Storage	213.9 GE	28.8...	13.5%

Narration

This shows me that my VM is consuming Resources that we call **Commodities**, like memory and CPU from the underlying host.

In this example I am consuming a **Commodity** that's called memory from my host, in this case 8.8GB.

And what you can see here constitutes some sort of shopping list that contains all resources that I need, like amount of memory, this amount of CPU, this amount of storage, and so on and so forth.

And Turbonomic is then trying to optimize the virtual price of this shopping list while guaranteeing the application performance.

So basically, and that's what I explained in the beginning before going into the demo, we're doing commodity market optimization applied to your infrastructure.

So as an example:

- If I have over provisioned memory, I will virtually sell some memory by scaling down, which will make my shopping list overall cheaper.
- If I see that I don't have enough memory, I will buy some memory, then my shopping list will get a little bit more expensive, but I'm ensuring my performance.

Turbo_897_Test

Commodity	Entity	Type	Capacity	Capacity Source	Used	Used Source	Utilization
Storage Amount	FROM: Turbo_897_Test Disk 2	Consumed	250 GB	Target	911.87 MB	Target	0.36%
Memory	FROM: ded-wdc06s004.na.cloud.techzone.ibm.com	Consumed	3 TB	Target	8.88 GB	Target	0.29%
CPU	FROM: ded-wdc06s004.na.cloud.techzone.ibm.com	Consumed	229.82 GHz	Target	433.87 MHz	Target	0.19%
Memory Allocation	FROM: Resources-dedicated-wdc06 KEY: VirtualDataCent	Consumed	23.45 TB	Target	8.88 GB	Target	0.04%
Memory Provisioned	FROM: ded-wdc06s004.na.cloud.techzone.ibm.com	Consumed	29.99 TB	Target	9 GB	Target	0.03%
CPU Allocation	FROM: Resources-dedicated-wdc06 KEY: VirtualDataCent	Consumed	1.77 THz	Target	433.87 MHz	Target	0.02%
Storage Access	FROM: Turbo_897_Test Disk 1	Consumed	50,000 IOPS	Target	7 IOPS	Target	0.01%
CPU Provisioned	FROM: ded-wdc06s004.na.cloud.techzone.ibm.com	Consumed	68.95 THz	Target	4.79 GHz	Target	0.01%
Net Throughput	FROM: ded-wdc06s004.na.cloud.techzone.ibm.com	Consumed	6.1 GB/s	Target	0 KB/s	Target	0%
Storage Access	FROM: Turbo_897_Test Disk 3	Consumed	50,000 IOPS	Target	0 IOPS	Target	0%
Storage Provisioned	FROM: Turbo_897_Test Configuration	Consumed	23.44 TB	Target	1.51 MB	Target	0%
Storage Latency	FROM: Turbo_897_Test Disk 3	Consumed	100 msec	Target	0 msec	Target	0%
IO Throughput	FROM: ded-wdc06s004.na.cloud.techzone.ibm.com	Consumed	1.46 GB/s	Target	28 KB/s	Target	0%
Storage Access	FROM: Turbo_897_Test Disk 2	Consumed	50,000 IOPS	Target	0 IOPS	Target	0%
Storage Latency	FROM: Turbo_897_Test Disk 1	Consumed	100 msec	Target	0 msec	Target	0%
Swapping	FROM: ded-wdc06s004.na.cloud.techzone.ibm.com	Consumed	5 MB/s	Target	0 B/s	Target	0%
Storage Latency	FROM: Turbo_897_Test Disk 2	Consumed	100 msec	Target	0 msec	Target	0%
Storage Amount	FROM: Turbo_897_Test Configuration	Consumed	23.44 TB	Target	1.51 MB	Target	0%
Ballooning	FROM: ded-wdc06s004.na.cloud.techzone.ibm.com	Consumed	3 TB	Target	0 KB	Target	0%

Action

Click anywhere in the Clickthrough demo, or scroll in the live demo to show the Provided resources

Turbo_897_Test

Commodity	Entity	Type	Capacity	Capacity Source	Used	Used Source	Utilization
Virtual Memory	—	Provided	9 GB	Target	2.73 GB	Calculation	30.34%
Virtual Storage	/boot	Provided	796.66 MB	Target	182.07 MB	Target	22.85%
Virtual Storage	/usr/local/bin	Provided	2.99 GB	Target	663.9 MB	Target	21.68%
Virtual Storage	/var/lib/containerd	Provided	213.9 GB	Target	28.87 GB	Target	13.5%
Virtual CPU	—	Provided	4.79 GHz	Target	433.87 MHz	Target	9.06%
Energy	—	Provided	5.57 Wh	Target	0.47 Wh	Target	8.45%
Virtual Storage	/	Provided	79.96 GB	Target	3.42 GB	Target	4.28%
Carbon Footprint	—	Provided	5.57 g	Target	0.18 g	Target	3.18%
Virtual Storage	/tmp	Provided	1.99 GB	Target	32.35 MB	Target	1.59%
Virtual Storage	/var	Provided	199.9 GB	Target	0.94 GB	Target	0.47%
Virtual Storage	/data/turbonomic	Provided	249.88 GB	Target	809.39 MB	Target	0.32%
Virtual Storage	/var/lib/dbs	Provided	799.61 GB	Target	73.43 MB	Target	0.01%
Storage Provisioned	FROM: Turbo_897_Test Disk 1	Consumed	1.45 TB	Target	1.45 TB	Target	100%
Storage Provisioned	FROM: Turbo_897_Test Disk 3	Consumed	1 TB	Target	1 TB	Target	100%
Storage Provisioned	FROM: Turbo_897_Test Disk 2	Consumed	250 GB	Target	250 GB	Target	100%
Storage Amount	FROM: Turbo_897_Test Disk 1	Consumed	1.45 TB	Target	39.77 GB	Target	2.67%
Storage Amount	FROM: Turbo_897_Test Disk 3	Consumed	1 TB	Target	16.37 GB	Target	1.6%
Storage Amount	FROM: Turbo_897_Test Disk 2	Consumed	250 GB	Target	911.87 MB	Target	0.36%
Memory	FROM: ded-wdc06s004.na.cloud.techzone.ibm.com	Consumed	3 TB	Target	8.88 GB	Target	0.29%

On the other hand, I see that this VM is providing for example **VirtualMemory**, **Storage** as well as **Energy** and **CarbonFootprint**.

So those resources/commodities are being consumed by what is running on this VM, which might be containers, a database or any other workload.

In this case the **VM** provides 9GB of **VirtualMemory** and 2.73GB are used.

So that's how the optimization works, and this generates what we call actions.

Action

Click the **cross** on the top right and Click **Show all** in the Actions section.

Narration

The actions make suggestions in order to ensure the best possible performance and efficiency in our environment. In the Action-Category column we can determine whether it's a performance or efficiency action.

Action

Click **Details** on the first action

The screenshot shows the Action Center interface with the following details:

Left Sidebar: Includes icons for ACTION, SEARCH, PLAN, PARK, PLACE, DASHBOARD, REPORTS, and SETTINGS. A red box highlights the 'ACTION' icon.

Top Bar: Shows 'Action Center (2)' and 'NEW VIEW' with a cross icon.

Table Headers: RESIZE, Virtual Machine Name, Risk, Resize Direction, Current Value, New Value, Resize Attribute, Action Category, and Action.

Table Data:

RESIZE	VIRTUAL MACHINE	RISK	RESIZE DIRECTION	CURRENT VALUE	NEW VALUE	RESIZE ATTRIBUTE	ACTION CATEGORY	ACTION
Resizing Actions (2)	VMEM Allocate 2 GB	VCPUs Reclaim 1 vCPU					PERFORMANCE	DETAILS
	Turbo_897_Test		Upsize	4 GB	6 GB	Capacity	PERFORMANCE	DETAILS
	Turbo_897_Test		Downsize	2 vCPU	1 vCPU	Capacity	EFFICIENCY	DETAILS

Bottom Right: Page navigation and 'Rows per page: 30' dropdown.

In the diagram, we observe that the memory utilization on the **VirtualMachine** has been consistently at its maximum over the last 30 days. With this **Performance Action**, our goal is to bring the memory utilization down to 70-75%, providing a buffer for potential memory spikes and ensuring optimal efficiency. In the resource impact section, you can see that we are scaling the memory capacity from 9GB to 13GB to achieve this objective.

The screenshot shows the Action Center interface. On the left, there's a sidebar with various navigation icons. The main area is titled "Action Details" and shows a "RESIZE" action for "Virtual Machines (2)". A graph titled "VMEM PERCENTILE AND AVG. UTILIZATION" displays memory usage over time, with a red box highlighting the projected utilization line. To the right, sections for "ACTION ESSENTIALS" (State: Action can be accepted and executed immediately), "ENTITY DETAILS" (Name: Turbo_897_Test, Id: vm-7498, Target Name: dedwdc06-vc.na.cloud.techzone.ibm.com, Target Type: vCenter, Tags: Owner Email IBM: dennis.prevost.jr@ibm.com, Suspend Date: 2023.12.06), and "RESOURCE IMPACT" (CPU Ready, Utilization: 0.6% to 0.6%; VMem Capacity: 4 GB to 6 GB; VCPU Capacity: 4.79 GHz to 2.39 GHz; Virtual Storage Capacity: 2.99 GB to 79.96 GB; Virtual Storage Utilization: 21.7% to 4.3%) are visible. A green "EXECUTE ACTION" button is at the bottom.

Action

Click the **right arrow** on the top right to go to the second action

Narration

With this **Efficiency Action**, we will scale down from 2 to 1 CPU to increase the CPU utilization from ~30% to ~70% on this VM.

All the actions can be executed manually using the execute button.

The idea is to start with actions that are validated and executed manually. Then, we review the results and their trustworthiness and as the confidence in the solution and the quality of the actions grows we can start automating them, until all of our infrastructure is on "autopilot", as I always say.

The place where we define all this is called **Policies**. There, we can define different behaviours for different types of actions and environments.

For instance:

- Non-disruptive actions in a Development environment can be executed every night.

- Disruptive actions in a Production Environment can only be executed during a maintenance window and with an additional sign-off in a tool like ServiceNow.

Action Center (2)

RESIZE

Virtual Machines (2)

Action Details

Resize down vCPU for Virtual Machine [Turbo_897_Test](#) from 2 to 1 vCPUs
Underutilized vCPU in Virtual Machine Turbo_897_Test

VCPU PERCENTILE AND AVG. UTILIZATION

VCPU Utilization is below 11% for 95% of the time over the 30 day observation period

Legend: ● vCPU Daily Avg. ⏷ Projected vCPU 95th Percentile
● vCPU 30 day 95th Percentile

ACTION ESSENTIALS

ENTITY DETAILS

RESOURCE IMPACT

	CURRENT	AFTER ACTIONS
CPU Ready, Utilization	0.6%	0.6%
VMem, Capacity	4 GB	6 GB
VMem, P95th Utilization	100%	66.67%
VCPUs, Capacity	4.79 GHz	2.39 GHz
VCPUs, P95th Utilization	11%	22%
Virtual Storage, Capacity	2.99 GB	79.96 GB
Virtual Storage, Utilization	21.7%	4.3%

CONFIGURATION IMPACT

	CURRENT	AFTER ACTIONS
VCPUs, Capacity	2	1
Sockets, Capacity	2	1
CoresPerSocket, Capacity	1	1

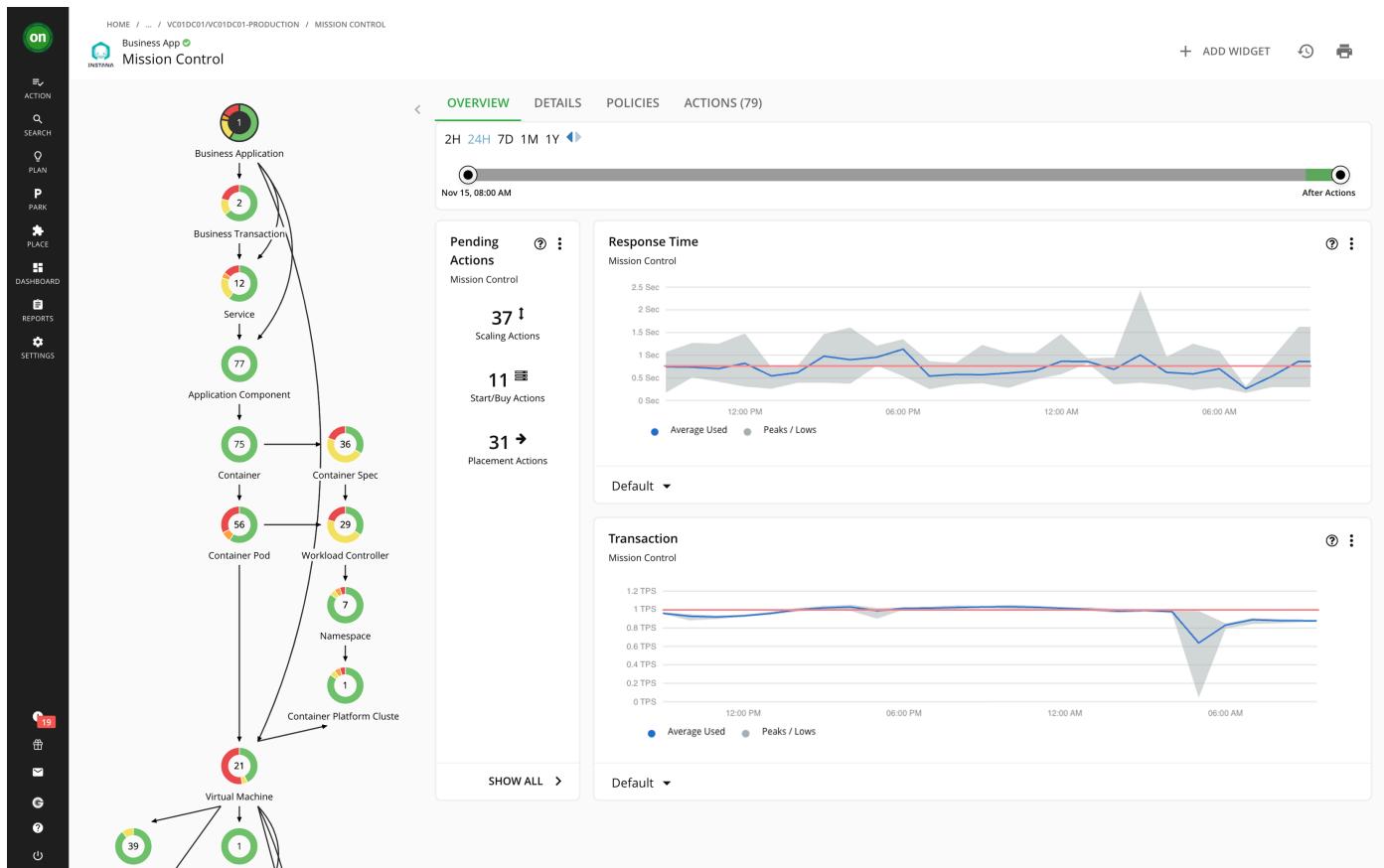
EXECUTE ACTION

2.6 Zoom into Business Application

Action

Click **Turbonomic**

Click **Mission Control** in the Top Business Application section



Narration

Let's refocus on the initial scenario and dive into my Application.

What I can do now is drill down into my **BusinessApplication**, called Mission Control.

I can observe the response times and transactions per second that are coming in from Instana, which serves as an Application Performance Monitoring (APM) tool—this could be Instana, Dynatrace, NewRelic, or others.

While not mandatory, it provides additional signals for addressing performance issues.

I've established my Service Level Agreement (SLA) indicated in red. Upon examination, I notice that the response time frequently exceeds the SLA, and the transactions per second are significantly below my expected rate of one transaction per second. Hence, there definitely is a performance issue. Let's now explore the suggested actions to remediate the problem.

2.7 Drill-down Actions

Action

Click **Actions** on the top bar

Click **Virtual Machines** in the **Resize** category on the left bar

The screenshot shows the Instana Mission Control interface. On the left, there's a sidebar with various icons and sections: ACTION, SEARCH, PLAN, PLACE, DASHBOARD, REPORTS, and SETTINGS. Below the sidebar, there are several small circular icons. The main area has a header with 'HOME / ... / VC01DC01/VC01DC01-PRODUCTION / MISSION CONTROL' and 'Business App Mission Control'. The top navigation bar includes 'OVERVIEW', 'DETAILS', 'POLICIES', and 'ACTIONS (79)', with 'ACTIONS (79)' being the active tab. A sub-header 'Resize Actions (10)' is followed by 'VMEM Allocate 10 GB'. To the right are buttons for '+ ADD WIDGET', 'EXECUTE ACTIONS', and other controls. The main content is a table with columns: 'Virtual Machine Name', 'Risk', 'Resize Direction', 'Current Value', 'New Value', 'Resize Attribute', 'Action Category', and 'Action'. The table lists 10 entries, all of which are 'vc01-centos-C06' with 'Virtual Memory' and 'Upsize' as the direction. The 'Action Category' column shows 'PERFORMANCE' for all rows. Each row has a 'DETAILS' button.

Virtual Machine Name	Risk	Resize Direction	Current Value	New Value	Resize Attribute	Action Category	Action	
vc01-centos-C06		Virtual Memory	Upsize	1 GB	2 GB	Capacity	PERFORMANCE	DETAILS
vc01-centos-C02		Virtual Memory	Upsize	1 GB	2 GB	Capacity	PERFORMANCE	DETAILS
vc01-centos-C01		Virtual Memory	Upsize	1 GB	2 GB	Capacity	PERFORMANCE	DETAILS
vc01-centos-C04		Virtual Memory	Upsize	1 GB	2 GB	Capacity	PERFORMANCE	DETAILS
vc01-centos-C03		Virtual Memory	Upsize	1 GB	2 GB	Capacity	PERFORMANCE	DETAILS
vc01-centos-C07		Virtual Memory	Upsize	1 GB	2 GB	Capacity	PERFORMANCE	DETAILS
vc01-centos-RQ01		Virtual Memory	Upsize	1 GB	2 GB	Capacity	PERFORMANCE	DETAILS
vc01-centos-M29		Virtual Memory	Upsize	1 GB	2 GB	Capacity	PERFORMANCE	DETAILS
vc01-centos-M24		Virtual Memory	Upsize	1 GB	2 GB	Capacity	PERFORMANCE	DETAILS
vc01-centos-M21		Virtual Memory	Upsize	1 GB	2 GB	Capacity	PERFORMANCE	DETAILS

Narration

Here we can see all the actions that have been generated for our business application. On the left bar you can see the different action categories, like Resize, Move, Provision and Scale.

As you can see here we have several VM's which we need to resize from 1GB to 2GB to ensure continuous performance. Let's focus on resize actions for the vc01-centos-C06 virtual machine.

Action

Click **vc01-centos-C06**

Action Details

Resize up vMEM for Virtual Machine **vc01-centos-C06** from 1 GB to 2 GB
vMEM Congestion in Virtual Machine vc01-centos-C06

VMEM PERCENTILE AND AVG. UTILIZATION

ACTION ESSENTIALS

ENTITY DETAILS

Name	vc01-centos-C06
Id	vm-291
Target Name	vc01.na.cloud.techzone.ibm.com
Target Type	vCenter

RESOURCE IMPACT

CURRENT	AFTER ACTIONS
CPU Ready, Utilization	6%
VMem, Capacity	1 GB
VMem, P95th Utilization	93%
VCPU, Capacity	2.39 GHz
VCPU, P95th Utilization	75%
Virtual Storage, Capacity	0.99 GB
Virtual Storage, Utilization	23.9%

POLICIES

Name	Type
vm_move	Automation
VMa_placement_resize	Automation
Vpr	Automation
vm_resize_move_itog	Automation
VM_replacement_FH	Automation

EXECUTE ACTION

Narration

Here in the action details we see that we would move our 95th percentile usage from above 95% to under 50% with the 1GB increase. Obviously I can only scale in 1GB increments, so 2GB would be the nearest to achieve at most 70% of utilisation.

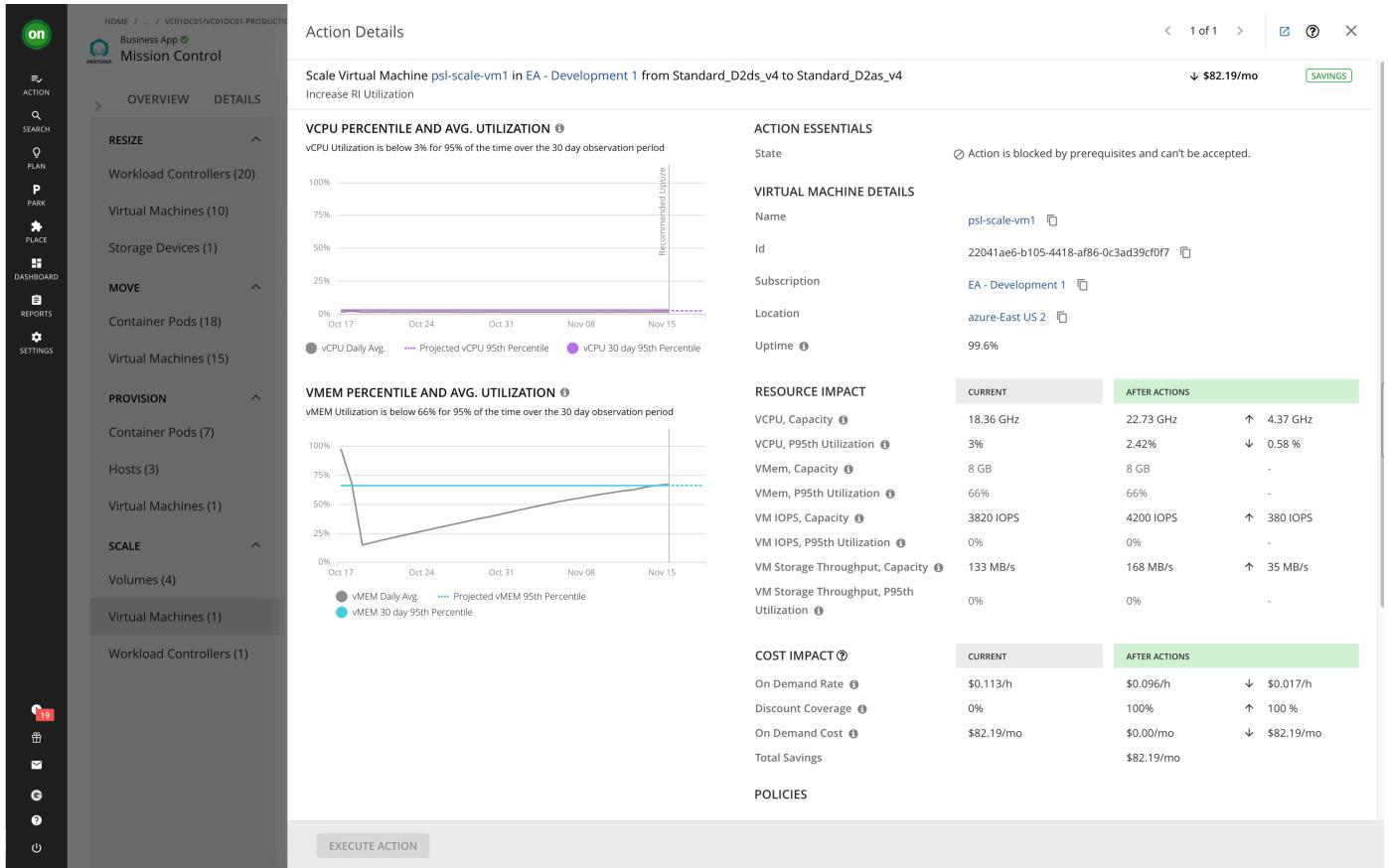
Next we will have a look at the scaling actions of the VM's.

Action

Click the **cross** on the top right

Click **Virtual Machines** in the **Scale** category of the left bar

Click **psl-scale-vm1**



The screenshot shows the Turbonomic interface with the following details:

- Action Details:** Scale Virtual Machine **psl-scale-vm1** in EA - Development 1 from Standard_D2ds_v4 to Standard_D2as_v4.
- VCPU PERCENTILE AND AVG. UTILIZATION:** vCPU Utilization is below 3% for 95% of the time over the 30 day observation period. Chart shows utilization from Oct 17 to Nov 15.
- VIRTUAL MACHINE DETAILS:** Name: psl-scale-vm1, ID: 22041ae6-b105-4418-af86-0c3ad39cf0f7, Subscription: EA - Development 1, Location: azure-East US 2, Uptime: 99.6%.
- RESOURCE IMPACT:** CURRENT vs AFTER ACTIONS for various metrics like VCPU Capacity, VMem Capacity, VM Storage Throughput, etc.
- COST IMPACT:** CURRENT vs AFTER ACTIONS for On Demand Rate, Discount Coverage, On Demand Cost, and Total Savings.
- POLICIES:** EXECUTE ACTION button.

Narration

This action is not related to performance improvements, however it allows me to save money while still ensuring the performance of my application.

In the details i can see that I get more VCPU capacity and better performance for less money. In total I would save 82\$ per month if I scale my VM to the suggested instance type because:

- This instance type is less expensive than the previous one and
- It is already covered by capacity that I have bought upfront - what we call **DiscountCoverage** in Turbonomic.

So those actions will help us ensure future performance of our application while saving on our cloud bill.

Action

Click the **cross** on the top right

Click **Host** in the provision category of the left bar

RESIZE ▾

- Workload Controllers (20)
- Virtual Machines (10)
- Storage Devices (1)

MOVE ▾

- Container Pods (18)
- Virtual Machines (15)

PROVISION ▾

- Container Pods (7)
- Hosts (3)
- Virtual Machines (1)

SCALE ▾

- Volumes (4)
- Virtual Machines (1)
- Workload Controllers (1)

Narration

We could go through the whole list but what we have mainly to do is to provision three new hosts to make sure that we can tackle the future resource needs of my application so that we can spread out the workloads over more hosts than we have today. Those three actions that you can see here indicate exactly that.

So by taking all the actions that I have shown you, I should be able to tackle the problems that have been mentioned by the CTO.

Now to ensure that we don't rapidly run into new problems, let's plan ahead a little bit for next year.

Let me explain how we can start planning for future scenarios.

2.8 Explain Plan Management

Action

[Click Plan](#)

[Click New Plan](#)

Select a Plan Type

CONTAINER PLATFORM

- Optimize Container Cluster
- Migrate Container Workloads

PUBLIC CLOUD

- Optimize Cloud
- Buy VM Reservations
- Migrate to Cloud

ON-PREM

- Optimize On-prem
- Add Virtual Machines
- Hardware Refresh
- Host Decommission
- Virtual Machine Migration
- Merge Clusters

All visible procedures

Narration

With the **Plan** feature I'm able to create "what-if" scenarios and simulate possible outcomes for the future.

Here we can see the different plan scenarios which we could simulate.

First I'm focusing on the **Optimize On-prem** plan type in order to stabilize the performance of the RobotShop Application in the long term.

Action

Click **Optimize On-prem**

Click **all-vCenter-VMs**

Select **the actions of your choice** to be included in the plan

Click **Run Plan**

Optimize On-prem 327

SCOPE: all-vCenter-VMs

RUN AGAIN

RESULTS OVERVIEW | **PLAN ACTIONS (1065)**

Plan Summary (all-vCenter-VMs)

	Current	After Plan	Difference	%
Containers	678	676	2	▼ 0.3 %
Container Pods	423	423	0	0 %
Virtual Machines	314	315	1	▲ 0.3 %
Hosts	13	16	3	▲ 23.1 %
Storage	8	11	3	▲ 37.5 %
CPU	496 Cores	502 Cores	6 Cores	▲ 1.2 %
Memory	15.12 TB	15.17 TB	48 GB	▲ 0.3 %
Storage Amount	52.35 TB	122.8 TB	70.4 TB	▲ 134.6 %
Host Density	24.2 : 1	19.7 : 1	4.5 : 1	▼ 18.6 %
Storage Density	39.3 : 1	28.6 : 1	10 : 1	▼ 27.2 %

Plan Actions (all-vCenter-VMs)

- Start/Buy Actions: 77
- Scaling Actions: 276
- Configuration Actions: 3
- Placement Actions: 603
- Stop Actions: 2
- Delete Actions: 104

Narration

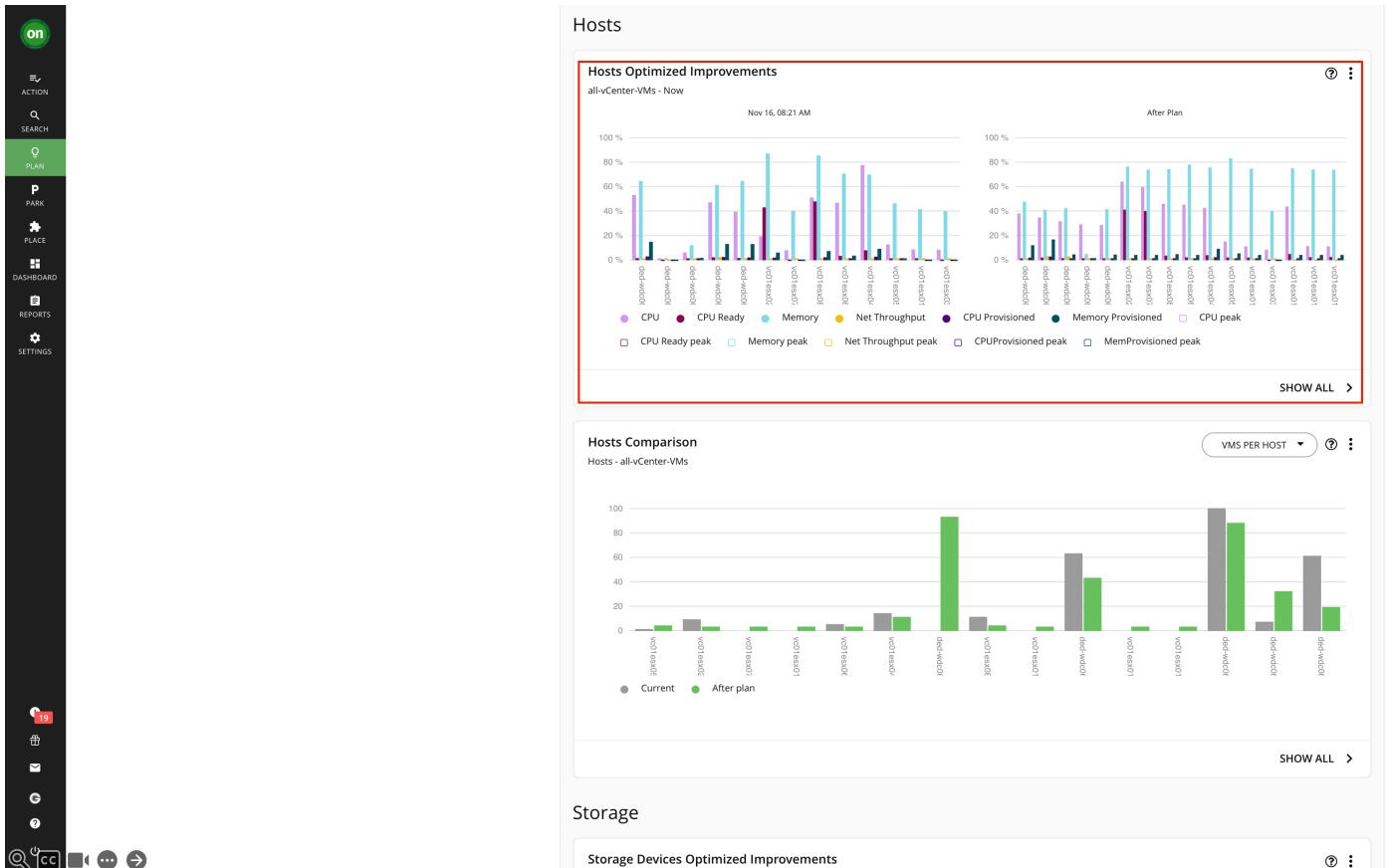
On this plan summary we get a good overview of the actions we have to implement to ensure smooth operations in the future:

- We will have to provision three new hosts (as we had already seen in the suggested actions)
- We will also have to provision more storage in order to ensure growth projected by Turbonomics machine learning algorithms

The goal is to lower the Host and Storage Density to make room for future needs.

Action

Scroll down to **Hosts** (click on the lower part of the slide for the click-through demo)



Narration

It's interesting to see that after the plan's execution we will have much more evenly distributed workloads on the different hosts than before.

For the second half of next year we plan to migrate the on-prem instances to cloud in order to reduce the footprint in our local data center.

Let me create a plan to help me get a good overview on how we could do that.

Action

Click **Plan**

Click **New Plan**

Click **Migrate to Cloud**

Click **Select all VM's**

Click **Azure Subscriptions**

Click **Pay as you go**

Click **Azure-Switzerland-North**

Click **Next: Licensing**

Click **Include OS cost**

Click **Next: Discount Settings**

Select **3 Years**

Click **Run Plan**

The screenshot shows the Turbonomic interface for migrating VMs to the cloud. On the left, a sidebar includes icons for Action, Search, Plan, Park, Place, Dashboard, Reports, and Settings. The main area has tabs for 'WHAT TO MIGRATE' (vc01-centos-RQ02, vc01-centos-M08...) and 'WHERE TO MIGRATE' (Pay-As-You-Go - Product Management...). Buttons for 'UPLOAD' and 'RUN AGAIN' are at the top right.

CONFIGURATION

- 1 Licensing
 - Include OS cost
- Discount Settings
 - Azure: Term (3 YEARS)
 - Discount Inventory (0 OUT OF 0 ACTIVE)
- Migrate 20 Virtual Machines
 - vc01-centos-C06 (Migrated to Pay-As-You-Go - Product Management/azur...)
 - vc01-centos-C02 (Migrated to Pay-As-You-Go - Product Management/azur...)
 - vc01-centos-M24 (Migrated to Pay-As-You-Go - Product Management/azur...)
 - vc01-centos-M07 (Migrated to Pay-As-You-Go - Product Management/azur...)
 - vc01-centos-M29 (Migrated to Pay-As-You-Go - Product Management/azur...)
 - vc01-centos-M18 (Migrated to Pay-As-You-Go - Product Management/azur...)
 - vc01-centos-RQ02 (Migrated to Pay-As-You-Go - Product Management/azur...)
 - vc01-centos-M30 (Migrated to Pay-As-You-Go - Product Management/azur...)

RESULTS OVERVIEW

Cloud Cost Comparison

	LIFT & SHIFT	OPTIMIZED	DIFFERENCE	%
Virtual Machines with performance risks	18 Out of 20	0 Out of 20	\$1,229.00 /mo	-54.4 %
Virtual Machines with efficiency opportunities	2 Out of 20	0 Out of 20	\$174.00 /mo	-30.5 %
On-Demand Compute Cost	\$2,258.00 /mo	\$1,029.00 /mo	-\$1,229.00 /mo	-54.4 %
Reserved Compute Cost	\$0.00 /mo	\$0.00 /mo	\$0.00 /mo	0 %
Storage Cost	\$174.00 /mo	\$121.00 /mo	-\$53.00 /mo	-30.5 %
Total Cost	\$2,432.00 /mo	\$1,150.00 /mo	-\$1,282.00 /mo	-52.7 %

Virtual Machine Mapping

LIFT & SHIFT

20 Virtual Machines
18 Standard_F1s
2 Standard_D16as_v5

OPTIMIZED

20 Virtual Machines
14 Standard_B1ms
3 Standard_F2s_v2
2 Standard_F1
1 Standard_A8m_v2

SHOW CHANGES >

Narration

I select all the VMs that I want to migrate. You probably remember that those were the ones that we had to scale up before.

So I select all the VMs that I want to migrate to cloud, then I select the Azure subscription account that I want to use and the region. Switzerland in this case as we want to keep the data in-country.

I will include the OS costs and I will pay three years upfront to buy reserved instances and I don't have any negotiated discounts available as of today.

So if I had already bought some discounts, they would show up here and I could use them if needed, but I don't have any in this case.

The result of the simulation contains two columns.

The first one is a simple **Lift and Shift**:

So if I take my VMs and my data center resources and just put them into the cloud, it would cost me about \$2,400 a month to migrate those 20 virtual machines of my RobotShop application into the cloud for about \$2,200 for compute cost or \$174 for storage cost.

Now let's look at the **Optimized** column:

If I optimize the resources before moving, I would have only around \$1,000 compute cost and \$120 of storage cost for a total cost of \$1,150 a month for the migration project.

And thanks to Turbonomic, I can gain about \$1,300. This doesn't seem like an enormous amount but 50% does.

So if you scale this to your cloud cost, imagine what a 25%-50% reduction in cost would mean to you.

Furthermore I can upload those migration plans to the Azure Migration Assistant to help with the migration.

And with that I am confident that the immediate performance problems are being solved and that we have a solid plan to stabilize the platform for the next 6 months and a good idea for the subsequent move to the cloud.

So Christmas and the holiday selling season are saved and my CTO should be happy, at least until next year, when we start talking about moving to cloud.

Demonstration summary

Narration

The Turbonomic demonstration focused on addressing critical performance issues within the RobotShop application. It started with an introduction to Turbonomic's key terminologies, followed by a step-by-step flow covering logging in, understanding the SupplyChain, integrating new targets (RobotShop), explaining commodities and optimization, and exploring actions within the Business Application.

I highlighted Turbonomic's ability to intelligently correlate data from various sources and its role as a centralized hub for managing and optimizing resources. Key features, such as SupplyChain modeling, target integration, and commodities optimization, were demonstrated to emphasize the platform's holistic approach.

Drilling down into specific actions, both for resizing and scaling virtual machines, illustrated Turbonomic's granular insights and recommendations. The Plan Management feature showcased its utility in planning for future scenarios, optimizing on-premises resources, and simulating cloud migration.

In summary, the demonstration successfully showcased Turbonomic's capabilities in providing actionable insights, automating recommendations, and optimizing resource management. The presenter emphasized the platform's role in addressing immediate performance issues, ensuring customer satisfaction, and planning for future scalability and cloud migration.

Thank you very much for listening!