



DISCOVERING AND MANAGING HOSTS WITH IMPROVED ZABBIX HOST PROTOTYPES IN ZABBIX 6.2

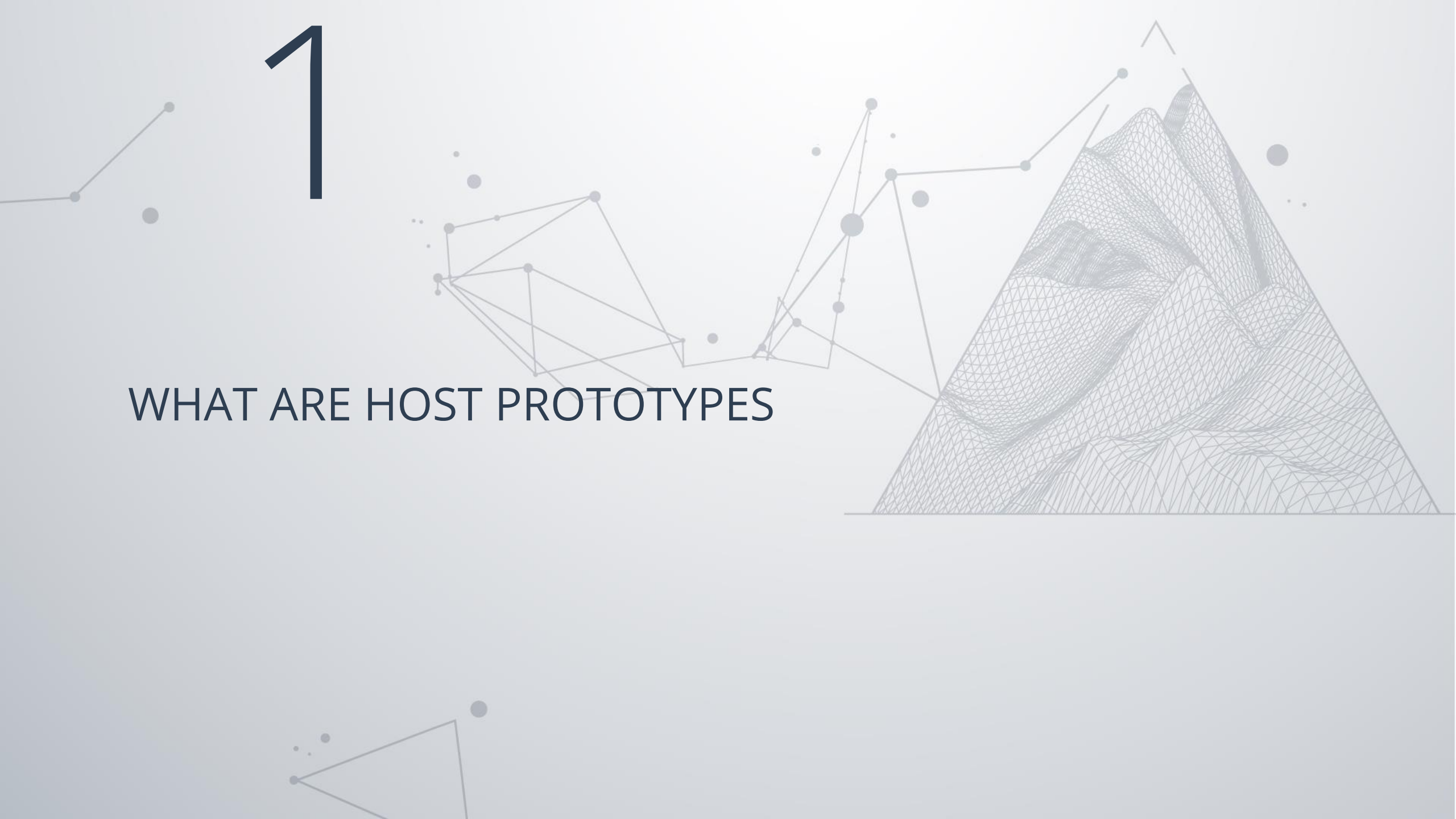


Kaspars Mednis
Chief trainer



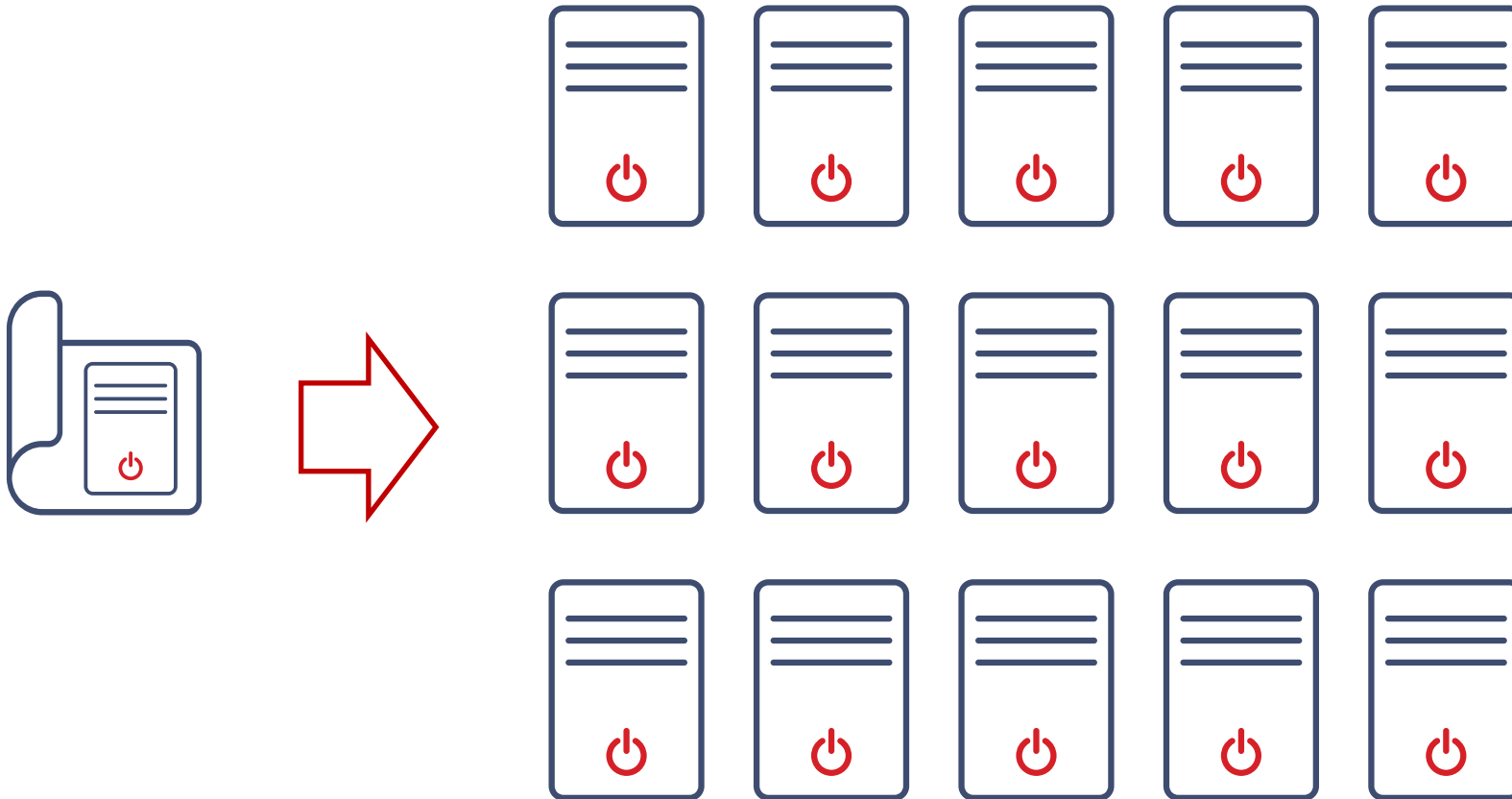
1

WHAT ARE HOST PROTOTYPES

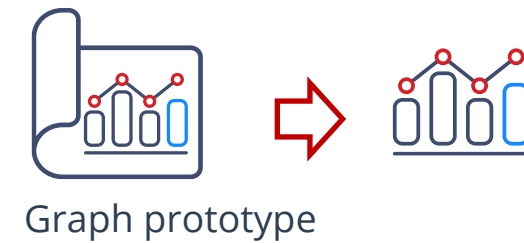
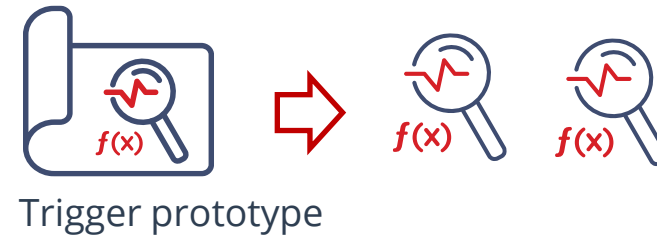
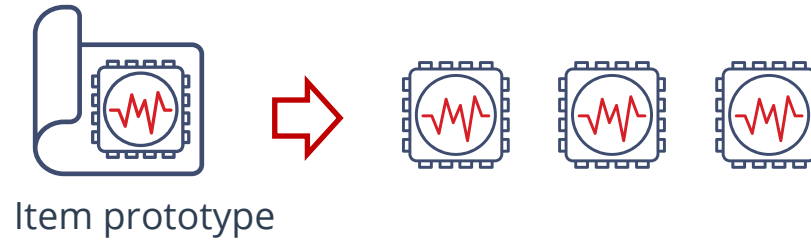


WHAT ARE HOST PROTOTYPES?

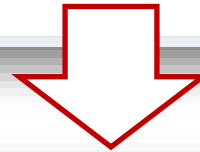
- ✓ Host prototypes are **blueprints** for hosts created by Low Level Discovery rules



HOW HOST PROTOTYPES **WORK?**



<input type="checkbox"/>	Template	Name ▲	Items	Triggers	Graphs	Hosts	Key	Interval	Type	Status
<input type="checkbox"/>	VMware	Discover VMware clusters	Item prototypes 1	Trigger prototypes	Graph prototypes	Host prototypes	vmware.cluster.discovery[{\$VMWARE.URL}]	1h	Simple check	Enabled
<input type="checkbox"/>	VMware	Discover VMware datastores	Item prototypes 4	Trigger prototypes	Graph prototypes	Host prototypes	vmware.datastore.discovery[{\$VMWARE.URL}]	1h	Simple check	Enabled
<input type="checkbox"/>	VMware	Discover VMware hypervisors	Item prototypes	Trigger prototypes	Graph prototypes	Host prototypes 1	vmware.hv.discovery[{\$VMWARE.URL}]	1h	Simple check	Enabled
<input type="checkbox"/>	VMware	Discover VMware VMs	Item prototypes	Trigger prototypes	Graph prototypes	Host prototypes 1	vmware.vm.discovery[{\$VMWARE.URL}]	1h	Simple check	Enabled
Displaying 4 of 4 found										



Host

Tags

Macros 1

Inventory

Encryption

* Host name

{#VM.UUID}

Visible name

{#VM.NAME}

Templates

Name

VMware Guest

type here to search

Action

Unlink

Select

* Host groups

Applications x

type here to search

Select

Group prototypes

{#CLUSTER.NAME} (vm)

Remove

{#DATACENTER.NAME}/{#VM.FOLDER} (vm)

Remove

{#HV.NAME}

Remove

Add

Interfaces

Inherit

Custom

Type

IP address

DNS name

Connect to

Port

Default

Agent

{#VM.IP}

IP

DNS

10050

Remove

Add

HOW HOST PROTOTYPES ARE CREATED

- ✓ Low Level Discovery (LLD) Rule is **executed**, and JSON data structure is created
- ✓ Based on host prototype definitions new hosts are created **automatically**
- ✓ Hosts can also be **modified** or **deleted** based on LLD results

<input type="checkbox"/> Name ▲	Items	Triggers	Graphs	Discovery	Web	Interface	Proxy	Templates	Status	Availability	Agent encryption	Info	Tags
<input type="checkbox"/> Discover VMware hypervisors: 192.168.3.67	Items 32	Triggers 7	Graphs	Discovery 2	Web	192.168.3.67:10050		VMware Hypervisor	Enabled	ZBX	None		
<input type="checkbox"/> Discover VMware hypervisors: 192.168.3.80	Items 42	Triggers 9	Graphs	Discovery 2	Web	192.168.3.80:10050		VMware Hypervisor	Enabled	ZBX	None		
<input type="checkbox"/> Discover VMware VMs: centos7-amd64-zabbix-agent-build	Items 49	Triggers 1	Graphs	Discovery 3	Web	192.168.12.80:10050		VMware Guest	Enabled	ZBX	None		
<input type="checkbox"/> Discover VMware VMs: centos8-amd64-zabbix-agent-build	Items 49	Triggers 1	Graphs	Discovery 3	Web	192.168.12.78:10050		VMware Guest	Enabled	ZBX	None		
<input type="checkbox"/> Discover VMware VMs: kubernetes-node1	Items 54	Triggers 1	Graphs	Discovery 3	Web	192.168.7.213:10050		VMware Guest	Enabled	ZBX	None		
<input type="checkbox"/> Discover VMware VMs: kubernetes-node2	Items 54	Triggers 1	Graphs	Discovery 3	Web	192.168.7.214:10050		VMware Guest	Enabled	ZBX	None		
<input type="checkbox"/> Discover VMware VMs: kubernetes-node3	Items 54	Triggers 1	Graphs	Discovery 3	Web	192.168.7.215:10050		VMware Guest	Enabled	ZBX	None		
<input type="checkbox"/> Discover VMware VMs: kubernetes-node4	Items 58	Triggers 1	Graphs	Discovery 3	Web	192.168.7.216:10050		VMware Guest	Enabled	ZBX	None		
<input type="checkbox"/> Discover VMware VMs: logstash	Items 49	Triggers 1	Graphs	Discovery 3	Web	192.168.10.66:10050		VMware Guest	Enabled	ZBX	None		
<input type="checkbox"/> Discover VMware VMs: Oracle 19C	Items 49	Triggers 1	Graphs	Discovery 3	Web	192.168.10.67:10050		VMware Guest	Enabled	ZBX	None		
<input type="checkbox"/> Discover VMware VMs: Oracle 21C	Items 49	Triggers 1	Graphs	Discovery 3	Web	192.168.10.65:10050		VMware Guest	Enabled	ZBX	None		
<input type="checkbox"/> Discover VMware VMs: vcenter	Items 161	Triggers 1	Graphs	Discovery 3	Web	192.168.3.15:10050		VMware Guest	Enabled	ZBX	None		

Displaying 12 of 12 found

HOST EXAMPLE IN ZABBIX 6.2

Host

HostIPMITagsMacros 5InventoryEncryption

Discovered byDiscover VMware VMs

* Host name5035da99-5a57-97bf-4616-d522f698ff50

Visible namekubernetes-node1

Templates ?

Name	Action
VMware Guest (linked by host discovery)	
<input type="text" value="type here to search"/>	Select

* Host groups

(vm) x 192.168.3.80 x Applications x HQ/Kubernetes HA (vm) x

Select

Interfaces	Type	IP address	DNS name	Connect to	Port	Default
Agent		192.168.7.213		IPDNS	10050	<input checked="" type="radio"/>

Description

Monitored by proxy

(no proxy) v

Enabled

☒

Update

Clone

Full clone

Delete

Cancel

Read only

Can be changed

Read only

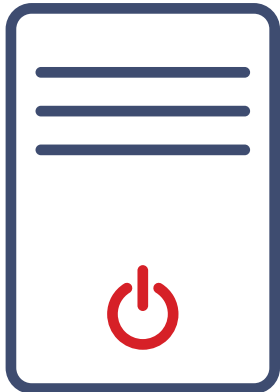
2

HOW HOST PROTOTYPES ARE
DEFINED



HOST PROPERTIES

- ✓ Host prototypes are defined inside **Low Level Discovery (LLD)** rules
- ✓ Common host properties like name, host groups and linked templates are defined **for all hosts** which will be created from this prototype



Host Tags Macros 1 Inventory Encryption

* Host name

Visible name

Templates	Name	Action
	VMware Guest	Unlink
	<input data-bbox="996 929 1730 968" type="text" value="type here to search"/>	Select

* Host groups [Select](#)

type here to search

Group prototypes

<input data-bbox="996 1086 1730 1125" type="text" value="{#CLUSTER.NAME} (vm)"/>	Remove
<input data-bbox="996 1143 1730 1182" type="text" value="{#DATACENTER.NAME}/{#VM.FOLDER} (vm)"/>	Remove
<input data-bbox="996 1200 1730 1239" type="text" value="{#HV.NAME}"/>	Remove

[Add](#)

WHAT IS LLD MACRO?

- ☑ LLD macros are values extracted from the discovery data

vmware.vm.discovery		
Performs virtual machine discovery.	{#VM.UUID}	Unique virtual machine identifier.
	{#VM.ID}	Virtual machine identifier (VirtualMachine managed object name).
	{#VM.NAME}	Virtual machine name.
	{#HV.NAME}	Hypervisor name.
	{#HV.UUID}	Unique hypervisor identifier.
	{#HV.ID}	Hypervisor identifier (HostSystem managed object name).
	{#CLUSTER.NAME}	Cluster name, might be empty.
	{#DATACENTER.NAME}	Datacenter name.
	{#DATASTORE.NAME}	Datastore name.
	{#DATASTORE.UUID}	Datastore identifier.
	{#VM.IP}	Virtual machine IP address, might be empty.
	{#VM.DNS}	Virtual machine DNS name, might be empty.
	{#VM.GUESTFAMILY}	Guest virtual machine OS family, might be empty.
	{#VM.GUESTFULLNAME}	Full guest virtual machine OS name, might be empty.
	{#VM.FOLDER}	The chain of virtual machine parent folders, can be used as value for nested groups; folder names are combined with "/". Might be empty.
	{#VM.TOOLS.STATUS}	VMware virtual machine tools state.
	{#VM.POWERSTATE}	VMware virtual machine power state (poweredOFF, poweredOn, or suspended).
	{#VM.RPOOL.ID}	Resource pool identifier.
	{#VM.RPOOL.PATH}	Full resource pool path excluding the "root" name "Resources". Folder names are combined with "/".
	{#VM.SNAPSHOT.COUNT}	Number of VM snapshots.

HOW LLD WORKS

- ☑ LLD is based on structured data in **JSON format**

```
{  
  "{#HV.UUID}": "32383138-3830-5a43-3337-3238434d3645",  
  "{#HV.ID}": "host-603",  
  "{#HV.NAME}": "192.168.3.80",  
  "{#HV.IP}": "192.168.3.80",  
  "{#DATACENTER.NAME}": "HQ",  
  "{#CLUSTER.NAME}": "",  
  "{#PARENT.NAME}": "HQ",  
  "{#PARENT.TYPE}": "Datacenter",  
  "{#HV.NETNAME}": "mysqlldb.example.com"  
}, {  
  "{#HV.UUID}": "34353935-3836-435a-4338-3336344c4b4c",  
  "{#HV.ID}": "host-972",  
  "{#HV.NAME}": "192.168.3.67",  
  "{#HV.IP}": "192.168.3.67",  
  "{#DATACENTER.NAME}": "HQ",  
  "{#CLUSTER.NAME}": "",  
  "{#PARENT.NAME}": "HQ",  
  "{#PARENT.TYPE}": "Datacenter",  
  "{#HV.NETNAME}": "centos.example.com"  
}
```



ADDITIONAL HOST PROTOTYPE PROPERTIES

- ✓ Zabbix allows to define **User macros** (5.0) and **Tags** (5.2) for host prototypes
- ✓ Macro and tag values may contain **LLD macros** {#MACRO}

Host **Tags 2** Macros 1 Inventory Encryption

Name	Value	
environment	production	Remove
cluster	{#CLUSTER.NAME}	Remove

[Add](#)

Host **Tags 2** **Macros 1** Inventory Encryption

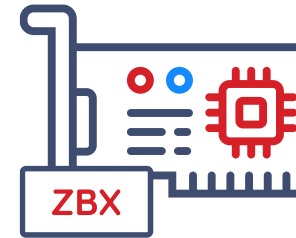
Host prototype macros Inherited and host prototype macros

Macro	Value	
{\$VMWARE.VM.UUID}	{#VM.UUID}	T v

[Add](#)

HOST INTERFACES

- ✓ **Parent host interface** was automatically inherited until Zabbix 5.2
- ✓ Now it is possible to choose **custom interface** using LLD macros
- ✓ **Any interface type** can be created (Agent, SNMP, JMX, IPMI)



Interfaces

Inherit

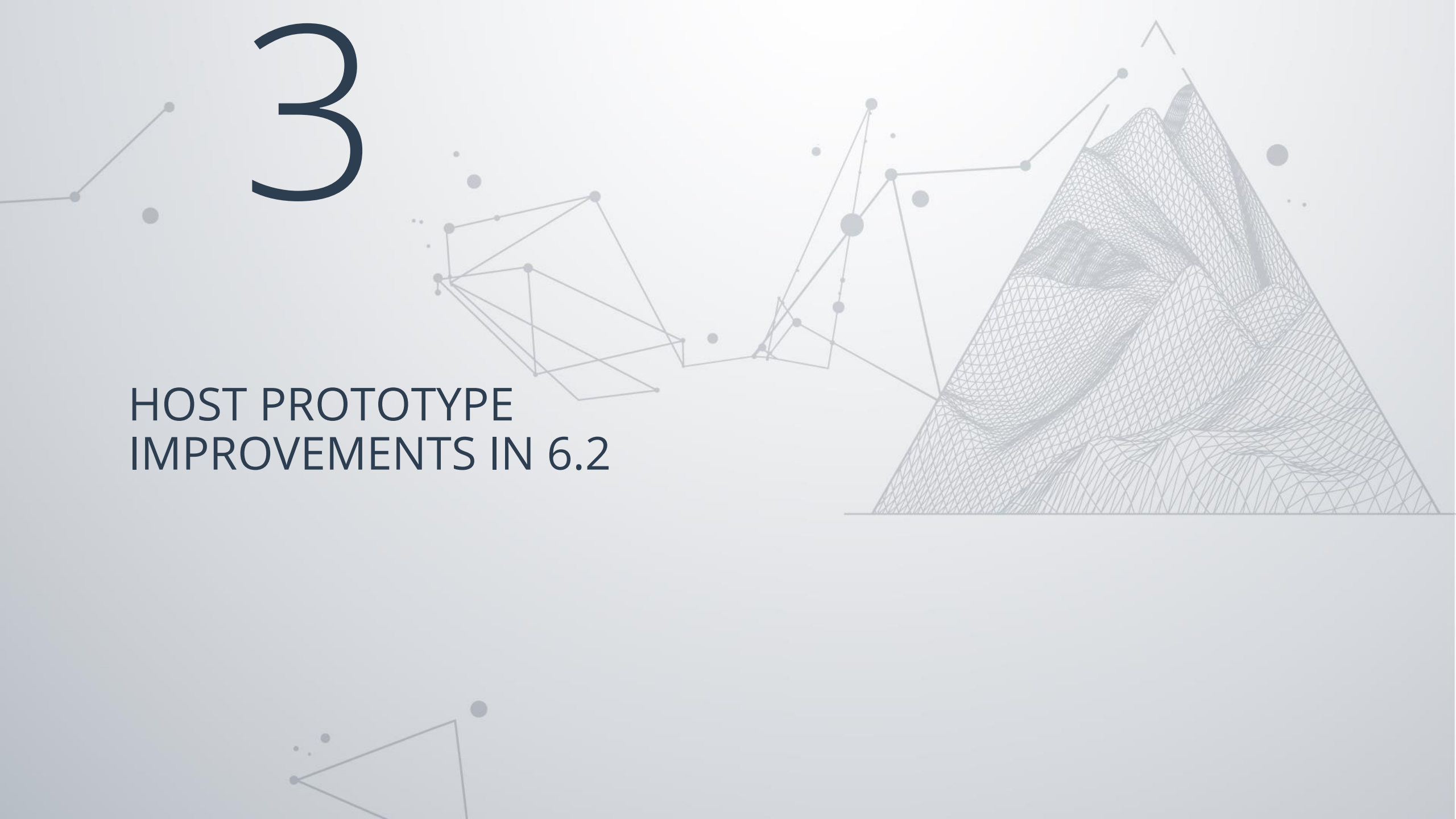
Custom

Type	IP address	DNS name	Connect to	Port	Default
Agent	{#VM.IP}		<div>IP</div> <div>DNS</div>	10050	<div><input checked="" type="radio"/></div> Remove

Add

3

HOST PROTOTYPE
IMPROVEMENTS IN 6.2



HOST PROPERTIES ON PREVIOUS ZABBIX VERSIONS

- ✓ Almost everything on the host created from prototype was **read only** until 6.2

Host Groups Templates IPMI Macros Inventory Encryption

* Groups Wireless Access Points ✕ Select

Group prototypes

{#APGROUP} Remove

[Add](#)

Host Groups Templates IPMI Macros Inventory Encryption

Linked templates

Name
Template Wireless Access Point

Host Groups Templates IPMI Macros Inventory Encryption

Host prototype macros Inherited and host prototype macros

Macro	Value	
{\$AP.LOCATION}	{#APLOCATION}	T ▼
{\$AP.NAME}	{#APNAME}	T ▼

MANUALY LINK TEMPLATES

- ✓ It is possible to manually link templates to hosts created from prototypes
- ✓ The VMWare virtual machine has Zabbix agent 2 installed in this example

Host

[Host](#) [IPMI](#) [Tags](#) [Macros 5](#) [Inventory](#) [Encryption](#)

Discovered by [Discover VMware VMs](#)

* Host name

5035c97e-e6bc-82f7-bfd2-3996ff911ce2

Visible name

ceph-node2

Templates ?

Name

VMware Guest (linked by host discovery)

ceph

Ceph by Zabbix agent 2

Action

Select

Select

* Host groups

Interfaces

Type	IP address	DNS name	Connect to	Port	Default
Agent	192.168.10.15		<div>IP</div> <div>DNS</div>	10050	<input checked="" type="radio"/>

MANUALLY ADD TAGS

- ✓ It is possible to **manually assign tags** to hosts created from prototypes
- ✓ The **environment** tag was created by LLD rule automatically
- ✓ **application** tag is added manually to the host later

Host

HostIPMI**Tags 2**Macros 5InventoryEncryption

Name	Value	
environment	vmware	Remove
application	ceph	Remove

[Add](#)







MANUALY ADD OR CHANGE USER MACROS

- ✓ A value of existing user macro can be **changed**
- ✓ A new user macro can be **added** if necessary

Host

Host IPMI Tags **Macros 5** Inventory Encryption

Host macros Inherited and host macros

Macro	Value		Description	
{VMWARE.PASSWORD}	 	VMware service {\$USERNAME} user password	Revert Remove (created by host discover)
{VMWARE.URL}	https://192.168.3.15/sdk/	T 	VMware service (vCenter or ESX hypervisor) SDK URL (https://servername/sdk)	Change Remove (created by host discover)
{VMWARE.USERNAME}	zabbix	T 	VMware service user name	Revert Remove (created by host discover)
{VMWARE.VM.UUID}	5035d352-f2c8-e5cb-ec64-28bc845f6f48	T 	UUID of guest virtual machine.	Change Remove (created by host discover)
{SMAX.PROCESSES}	650	T 	Maximum number of processes	Remove

[Add](#)

Update

Clone

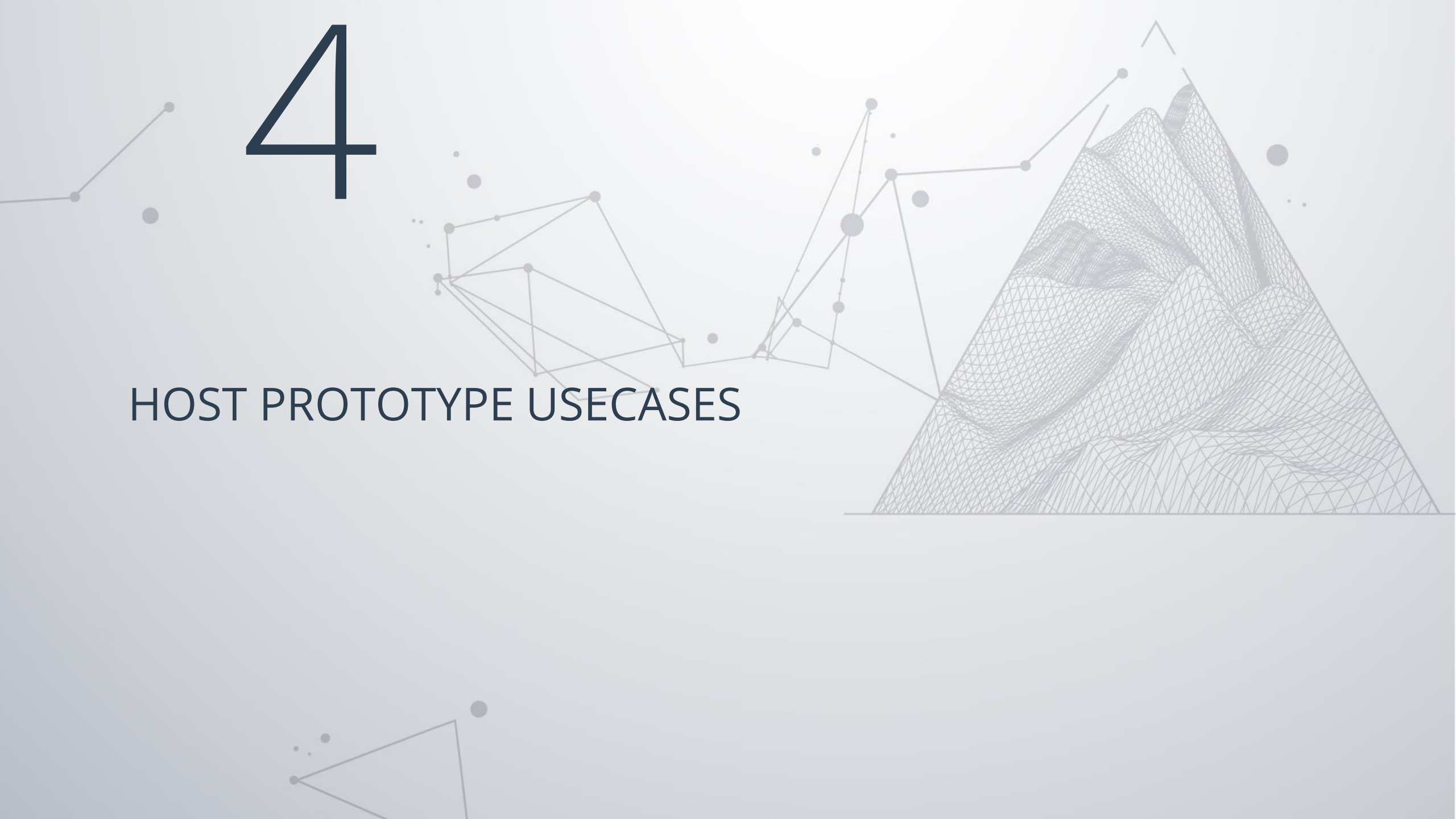
Full clone

Delete

Cancel

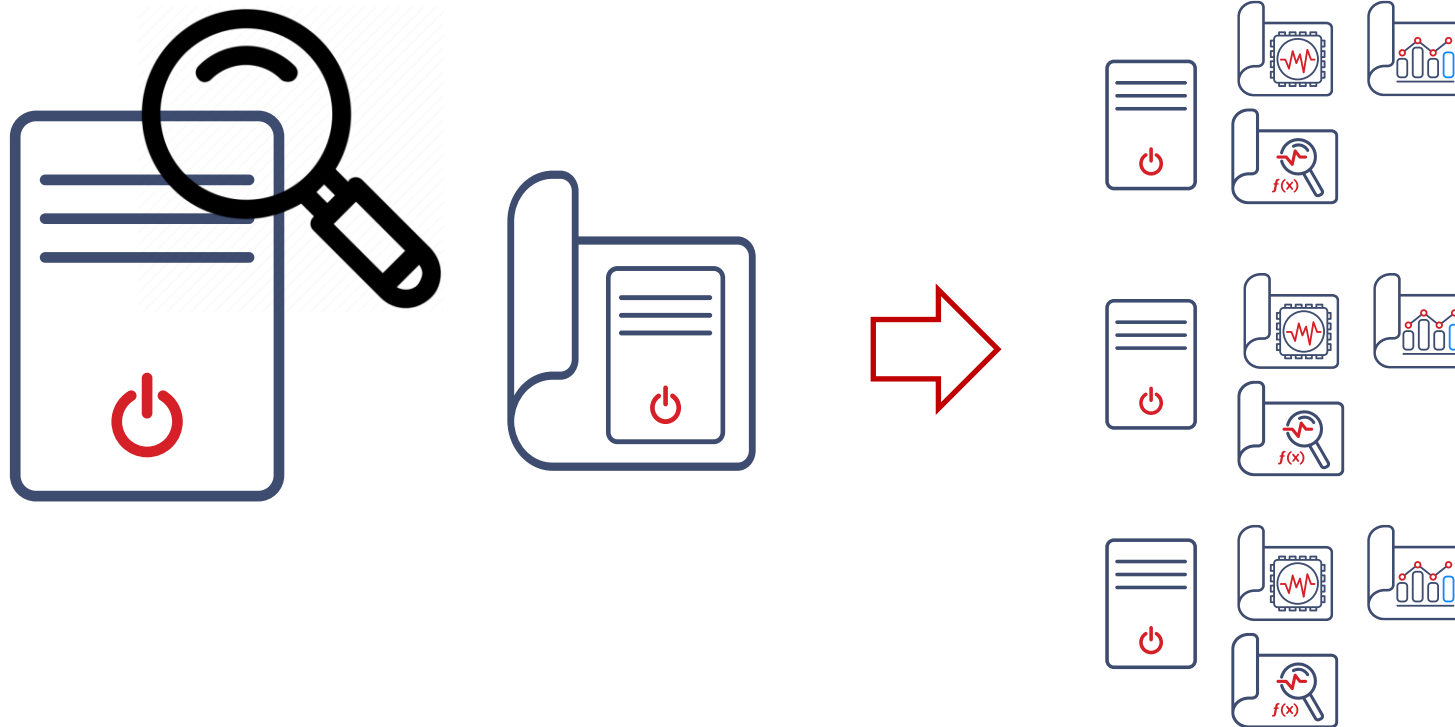
4

HOST PROTOTYPE USECASES



UNLEASH **THE FULL POWER** OF HOST PROTOTYPES

- ✓ Host prototypes may have **their own LLD rules** to discover metrics
- ✓ It is not possible to define host prototypes inside hosts created from prototypes



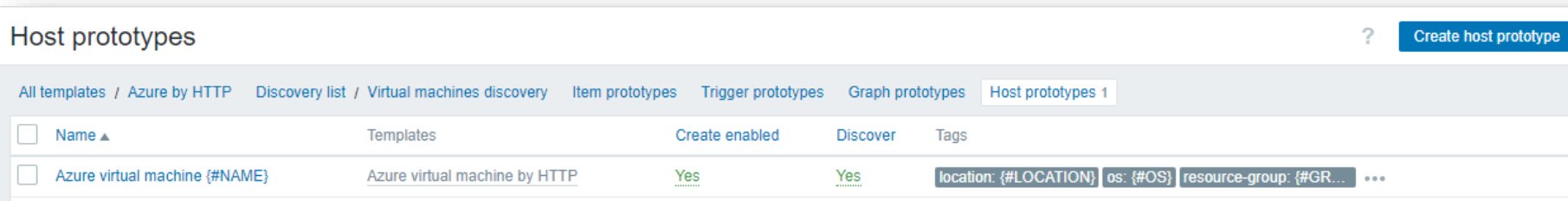
VMWARE MONITORING

- ✓ Monitor VMware infrastructure using built-in templates
- ✓ Automatically discover all hypervisors and virtual machines
- ✓ Monitor most important virtual machines using Zabbix agents
- ✓ Manually link templates to monitor critical application
- ✓ Tune settings using custom tags and user macros



CLOUD MACHINES MONITORING

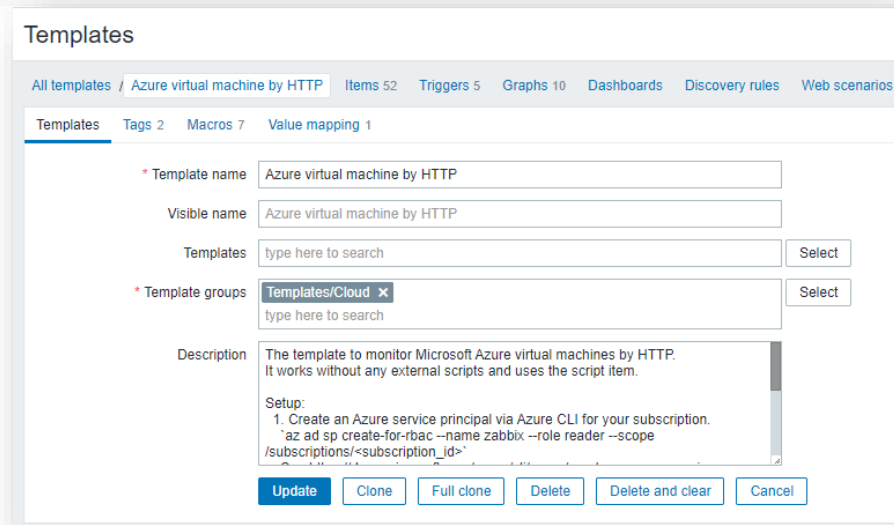
- ✓ Azure cloud provider is officially supported by Zabbix
- ✓ Use built-in template [Azure by HTTP](#)



The screenshot shows the 'Host prototypes' section in Zabbix. At the top, there's a 'Create host prototype' button. Below it, a breadcrumb trail reads: 'All templates / Azure by HTTP / Discovery list / Virtual machines discovery / Item prototypes / Trigger prototypes / Graph prototypes / Host prototypes 1'. A table lists the prototypes:

<input type="checkbox"/>	Name ▲	Templates	Create enabled	Discover	Tags
<input type="checkbox"/>	Azure virtual machine {#NAME}	Azure virtual machine by HTTP	Yes	Yes	location: {#LOCATION} os: {#OS} resource-group: {#GR...} ...

- ✓ [Azure virtual machine by HTTP](#) template will be automatically assigned to all hosts created from prototypes
- ✓ Link other templates if necessary



The screenshot shows the 'Templates' page in Zabbix for the 'Azure virtual machine by HTTP' template. The breadcrumb trail is: 'All templates / Azure virtual machine by HTTP / Items 52 / Triggers 5 / Graphs 10 / Dashboards / Discovery rules / Web scenarios'. The page has tabs for 'Templates', 'Tags 2', 'Macros 7', and 'Value mapping 1'. The 'Templates' tab is active, showing the following fields:

- * Template name: Azure virtual machine by HTTP
- Visible name: Azure virtual machine by HTTP
- Templates: type here to search [Select]
- * Template groups: Templates/Cloud [X] type here to search [Select]
- Description: The template to monitor Microsoft Azure virtual machines by HTTP. It works without any external scripts and uses the script item.

Below the description, there is a 'Setup' section with a list of instructions:

1. Create an Azure service principal via Azure CLI for your subscription.
`'az ad sp create-for-rbac --name zabbix --role reader --scope /subscriptions/<subscription_id>'`

At the bottom, there are buttons: 'Update', 'Clone', 'Full clone', 'Delete', 'Delete and clear', and 'Cancel'.

KUBERNETES MONITORING

- ✓ All **nodes** are discovered by LLD rule
- ✓ Linux by Zabbix agent template is automatically assigned to all discovered hosts

Discovery rules ? [Create discovery rule](#)

[All templates](#) / [Kubernetes nodes by HTTP](#) [Items 3](#) [Triggers 1](#) [Graphs](#) [Dashboards](#) [Discovery rules 3](#) [Web scenarios](#) [Filter](#)

<input type="checkbox"/>	Template	Name ▲	Items	Triggers	Graphs	Hosts	Key	Interval	Type	Status
<input type="checkbox"/>	Kubernetes nodes by HTTP	Node LLD: Cluster node discovery	Item prototypes	Trigger prototypes	Graph prototypes	Host prototypes 1	kube.node_host.discovery		Dependent item	Enabled
<input type="checkbox"/>	Kubernetes nodes by HTTP	Node LLD: Node discovery	Item prototypes 27	Trigger prototypes 15	Graph prototypes 1	Host prototypes	kube.node.discovery		Dependent item	Enabled
<input type="checkbox"/>	Kubernetes nodes by HTTP	Kubernetes: Get nodes: Pod discovery	Item prototypes							Enabled

[Host](#) [Tags 4](#) [Macros 2](#) [Inventory](#) [Encryption](#)

* Host name

Visible name

Templates

Name	Action
Linux by Zabbix agent	Unlink

[Select](#)

* Host groups

[Applications](#) [Select](#)

Group prototypes

[Remove](#)

[Add](#)

Interfaces [Inherit](#) [Custom](#)

Type	IP address	DNS name	Connect to	Port	Default
Agent	<input data-bbox="1166 1289 1416 1318" type="text" value="{#IP}"/>	<input data-bbox="1416 1289 1625 1318" type="text"/>	IP DNS	<input data-bbox="1735 1289 1842 1318" type="text" value="10050"/>	<input checked="" type="radio"/> Remove

- ✓ Kubeletes, Scheduler, Controller manager and API server are also monitored using host prototypes

USE HOST PROTOTYPES FOR OTHER PURPOSES

- ✔ Create hosts to logically group bulk data based on location or device by example
- ✔ 1000 hosts with 10 metrics can be easier to manage than one host with 10 000 items
- ✔ Access permissions can be assigned on host level
- ✔ Create dashboard for every host using templated dashboards
- ✔ Some use cases are HashiCorp consul nodes, wireless access points, databases, etc.



5

HOST PROTOTYPES WORKSHOP



ATTEND THE HOST PROTOTYPES **WORKSHOP**

- ✔ Workshop will be provided on Saturday 08:30-09:45
- ✔ Zabbix virtual machines will be provided by us
- ✔ An example scenario will show how to create hosts using host prototypes





THANK YOU!

