MANAGING TRIPLEO AND OPENSTACK WITH MANAGEIQ

Created by Ladislav Smola

CONTROL ALL THE THINGS

Automate, optimize, and control your cloud and virtualization services

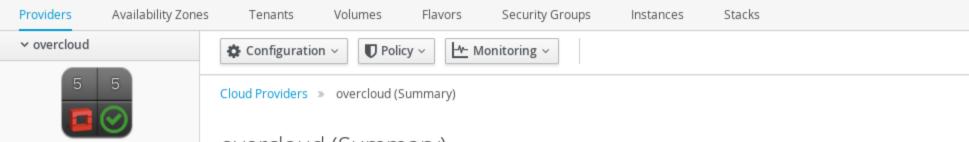
- Openstack Cloud in ManageIQ == TripleO Overcloud (or any end user cloud)
 - Openstack Infra in ManageIQ == TripleO Undercloud (RHEL OSP Director)

OPENSTACK CLOUD

- Inventory collection
- Capacity and utilization
- Smart state analysis(fleecing)
- Control and Policies
- Drift state
- Automate

INVENTORY COLLECTION

Provider Inventory



overcloud (Summary)

> Properties

> Relationships

Properties	
Host Name	172.16.23.10
Discovered IP Address	
Туре	OpenStack
API Port	5000
Management Engine GUID	0b14deb4-c906-11e5-8dc8-3c970e7e4812

Status	
Default Credentials	Valid
Last Refresh	Success - 25 Minutes Ago

Relationships	
Infrastructure Provider	anderclo
Availability zones	3 3
Cloud tenants	2 1
Flavors	8
Security groups	<u>2</u>
Instances	_ 5
Images	 5
Orchestration stacks	≣∘
Cloud volumes	■0

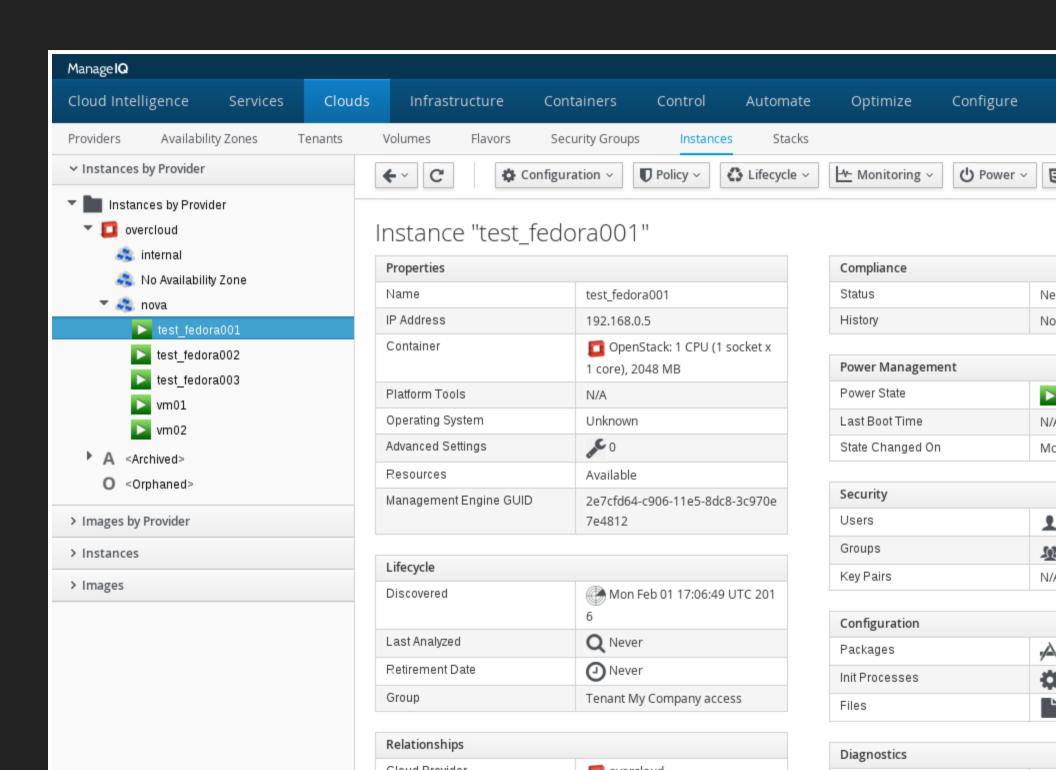
Smart Management	
Managed by Zone	a default
My Company Tags	No My Co

Properties	
Host Name	172.16.23.10
Discovered IP Address	
_	
Type	OpenStack
API Port	5000
Management Engine GUID	0b14deb4-c906-11e5-8dc8-3c970e7e4812

Status	
Default Credentials	Valid
Last Refresh	Success - 5 Minutes Ago

Relationships	
Infrastructure Provider	andercloud
Availability zones	♣ 3
Cloud tenants	♣ 1
Flavors	8
Security groups	2
Instances	₽ 5
Images	₽ 5
Orchestration stacks	≣ ∘
Cloud volumes	€0
Smart Management	
Managed by Zone	a default
My Company Tags	No My Company Tags have been assig ned

Instance Inventory



Cloud Provider	u overcioud
Infrastructure Provider	andercloud [1]
Deployment Role	遇 overcloud-Compute-54fnwnpz f34u

Running Processes

Event Logs

0

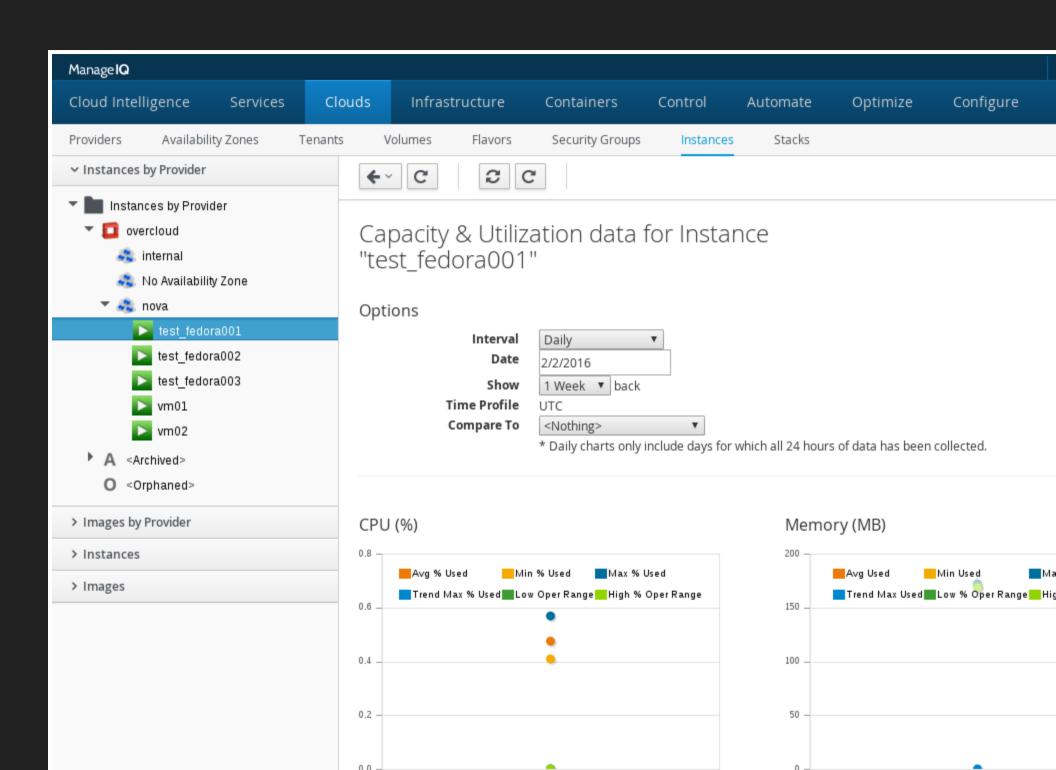
Smart Management

Properties	
Name	test_fedora001
IP Address	192.168.0.5
Container	OpenStack: 1 CPU (1 socket x 1 core), 2048 MB
Platform Tools	N/A
Operating System	Unknown
Advanced Settings	J € 0
Resources	Available
Management Engine GUID	2e7cfd64-c906-11e5-8dc8-3c970e 7e4812

Relationships	
Cloud Provider	overcloud
Infrastructure Provider	andercloud
Deployment Role	遇 overcloud-Compute-54fnwnpz f34u
Node	edfed5f4-7c80-468a-b33f-605 835aecc5a (NovaCompute)
Availability Zone	anova
Cloud Tenant	admin admin
Flavor	m1.small
VM Template	T Fedora22
Drift History	▲ None
Analysis History	Q None
Security Groups	<u>1</u>
Service	→ None
Stack	None

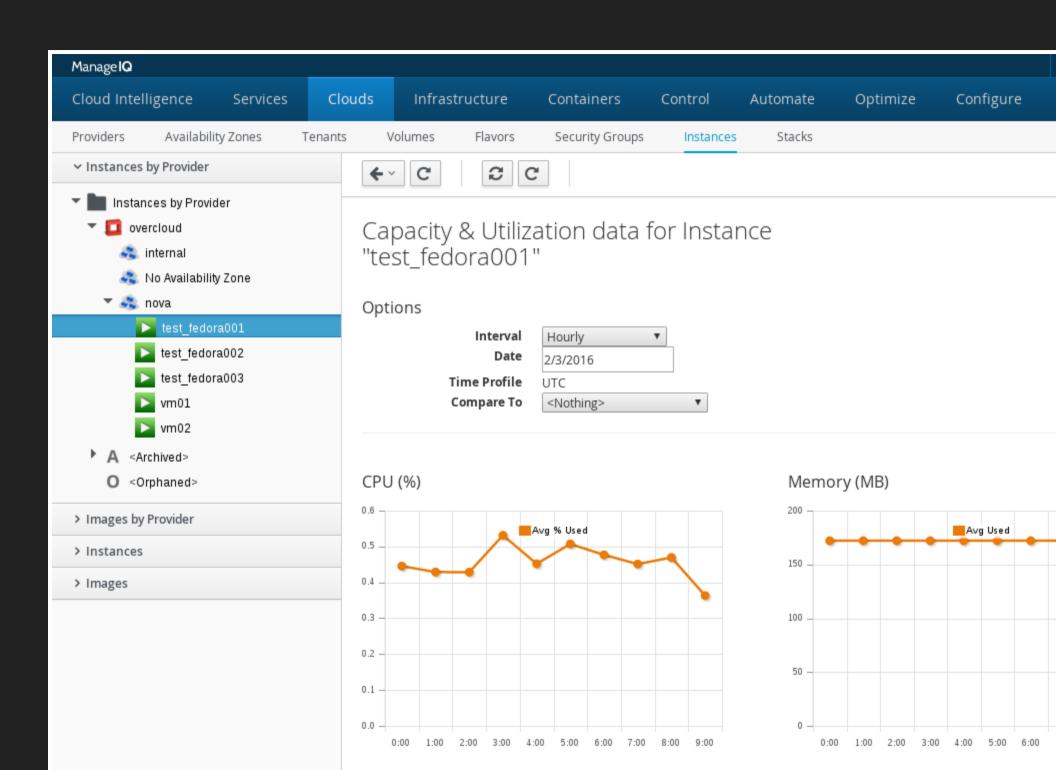
Smart state analysis attributes will be later in the presentation

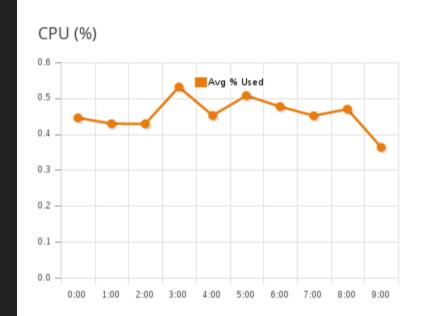
CAPACITY AND UTILIZATION



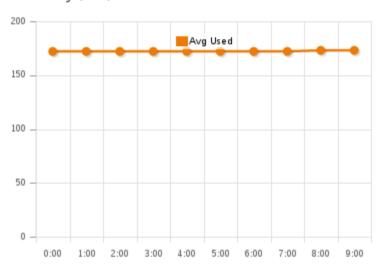
2/2

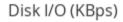
Disk I/O (KBps) Network I/O (KBps)

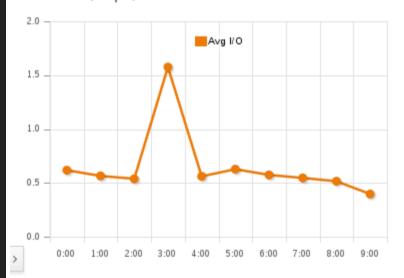




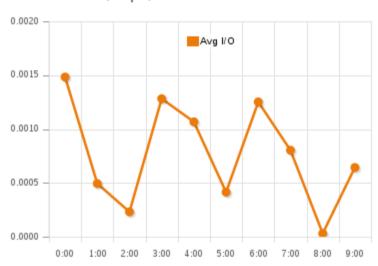








Network I/O (KBps)



SMART STATE ANALYSIS

In cloud, we don't have access to the VMs. We don't own the private keys.

- Snapshot the VM
- Mount the VM
- Perform Smart state analysis of the files
- Show users, groups, packages, init processes and files
- You need enough /tmp space in your appliance!

DRIFT STATE

Comparing entities and attributes we collect with their historical values. Examples later in the presentation.

CONTROL AND POLICIES

We can leverage all the entities we collect by Inventory collection and Fleecing

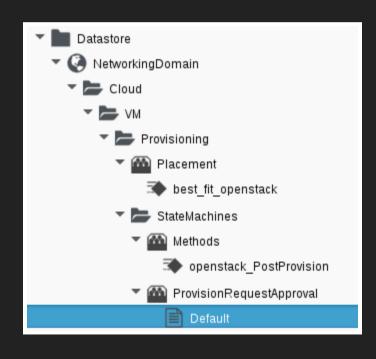
- Alerts notifications using email, SNMP traps, internal events
- Compliance pass/fail check, reporting (e.g. make sure none of my VMs suffer from the HeartBleed, ShellShock, etc.)
- Policy enforcement E.g. someone tries to clone a VM that is not allowed to clone, ManageIQ stops the action

AUTOMATE

AUTOMATE EXAMPLE SHOWING DEPLOYMENT OF THE VM

- In the UI, we fill what VMs we want to deploy, picking flavor, network, number of VMs, etc. Similar to Horizon.
- Deploying generates a request, that has to be approved.
- Approving invokes a provisioning workflow.
- After finishing provisioning workflow, we can invoke another workflow, e.g. Ansible Tower.

REQUEST APPROVAL WORKFLOW



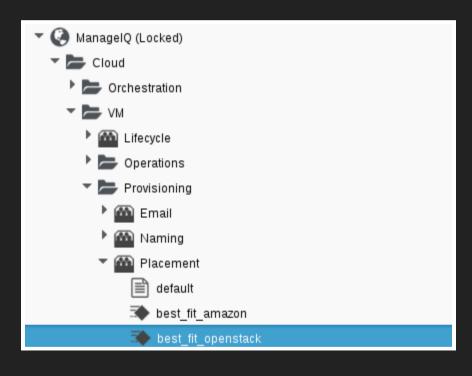
Name	Value	On Entry	On Exit	On Error	Collect
max_cpus					
max_vms	10				
max_memory					
max_retirement_days					
ॐ ⊘ ValidateRequest		validate_request		pending_request	
№		approve_request			

PROVISIONING WORKFLOW



Name		Value
₹ ⊙ (CustomizeRequest	/Cloud/VM/Provisioning/StateMachines/Methods/CustomizeRequest#\${/#miq_provision.source.vendor}
≥ ⊘ /	AcquireIPAddress	
≥ ⊘ /	AcquireMACAddress	
≥ ⊘ F	RegisterDNS	
≥ ⊘	RegisterCMDB	
≥ ⊘	RegisterAD	
≥ ⊘	Placement	/Cloud/VM/Provisioning/Placement/default#\${/#miq_provision.source.vendor}
≥ ⊘	PreProvision	/Cloud/VM/Provisioning/StateMachines/Methods/PreProvision#\${/#miq_provision.source.vendor}
≥ ⊘	Provision	/Cloud/VM/Provisioning/StateMachines/Methods/Provision
₹ ⊙ (CheckProvisioned	/Cloud/VM/Provisioning/StateMachines/Methods/CheckProvisioned
≥ ⊘	PostProvision	/Cloud/VM/Provisioning/StateMachines/Methods/PostProvision#\${/#miq_provision.source.vendor}
≥ ⊘	RegisterDHCP	
≥ ⊘ /	ActivateCMDB	
≥ ⊘ E	EmailOwner	/Cloud/VM/Provisioning/Email/MiqProvision_Complete?event=vm_provisioned
36 ⊘	Finished	/System/CommonMethods/StateMachineMethods/vm_provision_finished

Placement



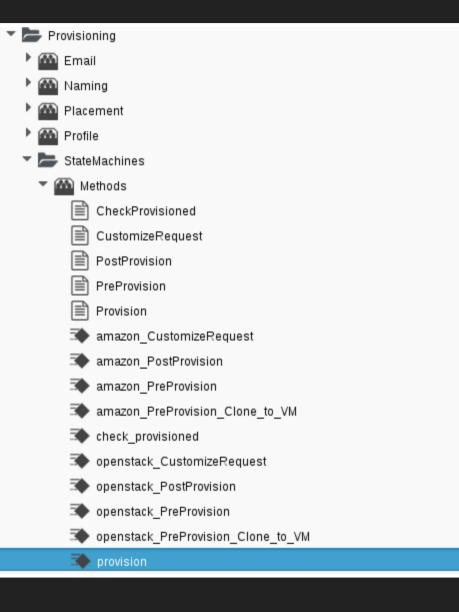
Domain Overrides (by priority)

NetworkingDomain

ManagelQ (Same Domain)

Data

Provision



Data

```
# Description: This method launches the provisioning job

# $evm.root["miq_provision"].execute
```

Post provision



Domain Overrides (by priority)

NetworkingDomain

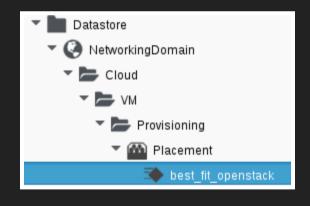
ManagelQ (Same Domain)

Data

```
# Description: This method is used to process tasks immediately after the VM has been provisioned
# Get provisioning object
prov = $evm.root["miq_provision"]

$evm.log("info", "Provisioning ID:<#{prov.id}> Provision Request ID:<#{prov.miq_provision_request.id}> PI
```

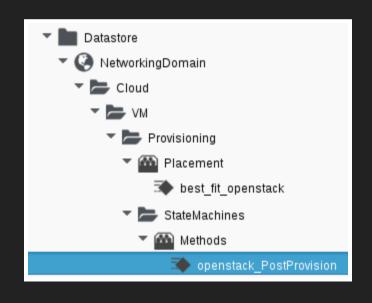
Override the Placement, to pick the most utilized private network



```
72 # Get variables
                    = $evm.root["mig provision"]
73 prov
74 image
                    = prov.vm template
75 raise "Image not specified" if image.nil?
77 tags = prov.get_tags
78 $evm.log("info", "Placement for tags: #{tags}")
80 case tags[:environment]
81
     when 'dev'
        # Get available private networks
       private networks = prov.eligible cloud_networks.select { |x| x.external_facing == false } # Filter only private networks that are connected to public network private networks = private_networks.select { |x| x.public_networks.count > 0 } # Sorted networks based on utilization, true parameter forces fresh API request, in the car # placement method as the led multiple times
83
84
85
86
87
        private_networks.sort_by! { |x| x.ip_address_left_count_live(true)}
88
        # Filter out full networks
89
        free_networks = private_networks.select { |x| x.ip_address_left_count_live > 0 }
# Pick a most utilized private network that is not full
        most utilized private network = free networks first
```

```
# If all networks are full, try to provision into first network, this should end with several tries (
        most utilized private network ]|= private networks.first
# GeT nice format of networks utilization. here we will just log it, but there can be e.g. calculation and sending email when it goes over some treshold.
 94
 96
 97
        network utilization nice format = {}
        private_networks.each { Tx| network utilization_nice_format[x.name] = x.ip_address_left_count_live } $evm.log("info", "Private Network number of free ip_addresses: #{network_utilization_nice_format}")
 98
 99
100
        if most_utilized_private_network.ip_address_left_count_live == 0
101
102
           # If utilization of least utilized network is IOO%. That means all networks are full
          $evm.log("warn", "All networks are full, trying to provision into full network.")
prov.set_option(:networks, [{:network_id => most_utilized_private_network.id}])
103
104
105
        else
106
          prov.set option(:networks, [{:network id => most utilized private network.id}])
           $evm.log("info", "Image=[#{image.name}] Cloud NeTwork=[#{most utilized private network.name}]")
107
108
        end
```

Override the Post provision, to retry when deployed to full private network



```
max retries = 3

if $\forall \text{evm.get} \text{ state_var("network_placement_retry_counter") > 3

raise "Couldn't place VM to any private network after #{max_retries} retries."

end
$\forall \text{evm.log("warn", "VM #{vm.name} was placed in full private network. Deleting VM and "Retry number #{$\forall \text{evm.get_state_var("network_placement_retry_counter} \text{ wm.destroy_if_failed} \text{ vm.destroy_if_failed} \text{ wm.coot['ae_next state to Placement" } \text{ $\forall \text{evm.root['ae_result'] = "Placement"} \text{ $\forall \text{evm.root['ae_result'] = 'restart'|} \text{ } \text{ $\forall \text{evm.root['ae_result'] = 'restart'|} \text{ } \text{
```

When deployment to private network succeeds, autoassociate floating ip

```
unless vm.floating_ip
# Get accessible public networks, meaning public networks are connected to this VM's p
public networks = vm.public networks
if public_networks.count > I

# This means that VM has two interfaces where each is connected to different public
# should not be appearing. If such architecture will appear, we need to specify whic
$evm.log("warn", "VM #{vm.name} is connected to more than one public network")
end

# Just pick connected public network, Vm should be connected to only one public networ
public_network = vm.public_networks.first

$evm.log("info", "Public Network IP address utilization in %: #{public_network.name}::
if public_network.ip_address_utilization >= 100
    # Optional action on full network.
end
```

```
begin

vm.associate floating_ip(public_network)

$evm.log("inTo", "Associating floating IP address from public network [#{public_network.name}] to

rescue StandardError => error

$evm.log("warn", "No more floating IPs left for VM #{vm.name} in public network #{public_network.r}

$evm.log("warn", "Attempting to delete unassigned Floating IPs and trying to associate floating IF

"in public network #{public network.name} for the second time.")

# Trying to clear free floating IPsOptionally filter what Floating IP you wish to keep

non_associated floating_ips = vm.ext management_system.floating_ips.select { |x| x.vm_id == nil }

vm.delete_floating_ips(non_associated_floating_ips)

# Trying to associate floating IP again

vm.associate_floating_ip(public_network)

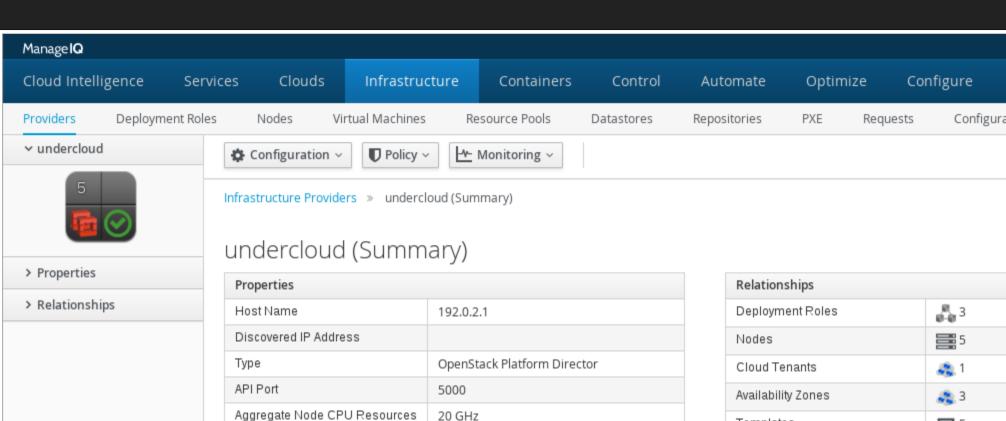
end
```

OPENSTACK INFRASTRUCTURE (RHEL OSP DIRECTOR)

- Inventory collection
- Smart state analysis
- Drift state
- Comparing nodes
- Autoscaling using automate

INVENTORY COLLECTION

Provider Inventory



Aggregate Node CPO Cores	10	
Management Engine GUID	abcf8bf4-c903-11e5-8dc8-3c970e7e4812	
Status		
AMQP Credentials	Valid	
Default Credentials	Valid	
Last Refresh	Success - About 1 Hour Ago	
States of Root Orchestration Stacks	overcloud status: UPDATE_COMPLETE	

50 GB

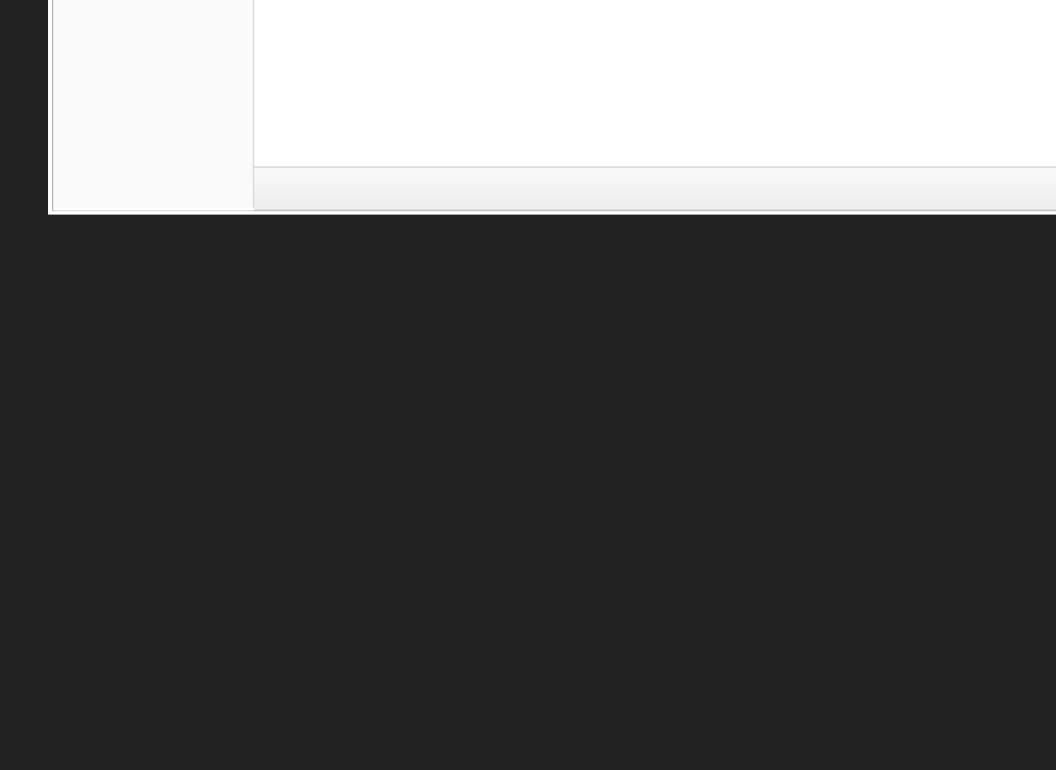
10

Aggregate Node Memory

Aggragata Nada CBU Caras

Aggregate Node CPUs

Nodes	5
Cloud Tenants	3 1
Availability Zones	3 3
Templates	 5
Orchestration stacks	■ 121
Smart Management	
Managed by Zone	a default
My Company Tags	No My C
	ned

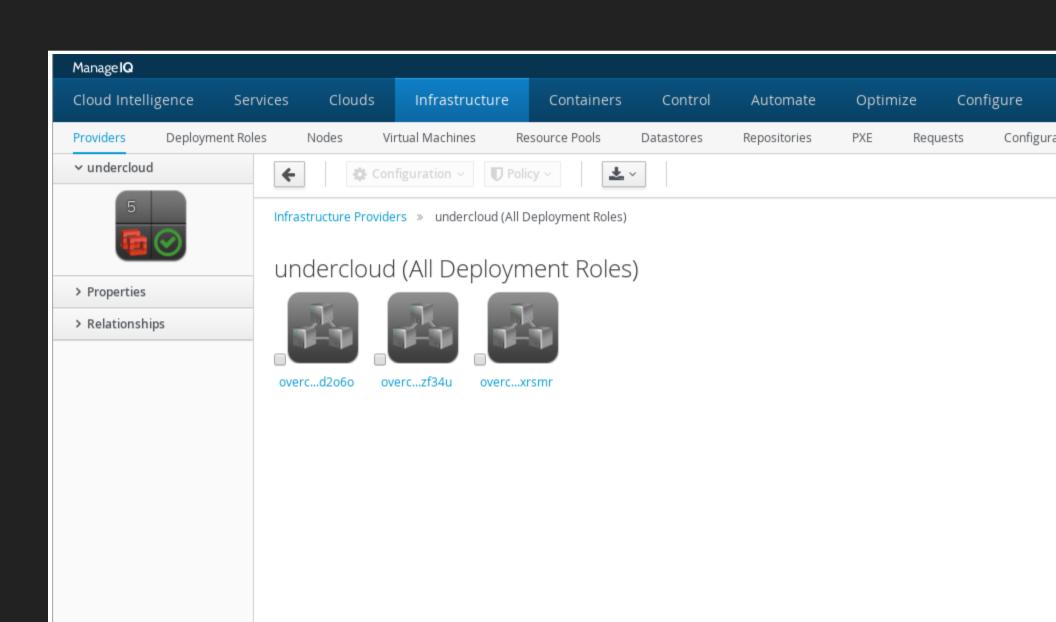


Properties	Properties	
Host Name	192.0.2.1	
Discovered IP Address		
Туре	OpenStack Platform Director	
API Port	5000	
Aggregate Node CPU Resources	20 GHz	
Aggregate Node Memory	50 GB	
Aggregate Node CPUs	10	
Aggregate Node CPU Cores	10	
Management Engine GUID	abcf8bf4-c903-11e5-8dc8-3c970e7e4812	

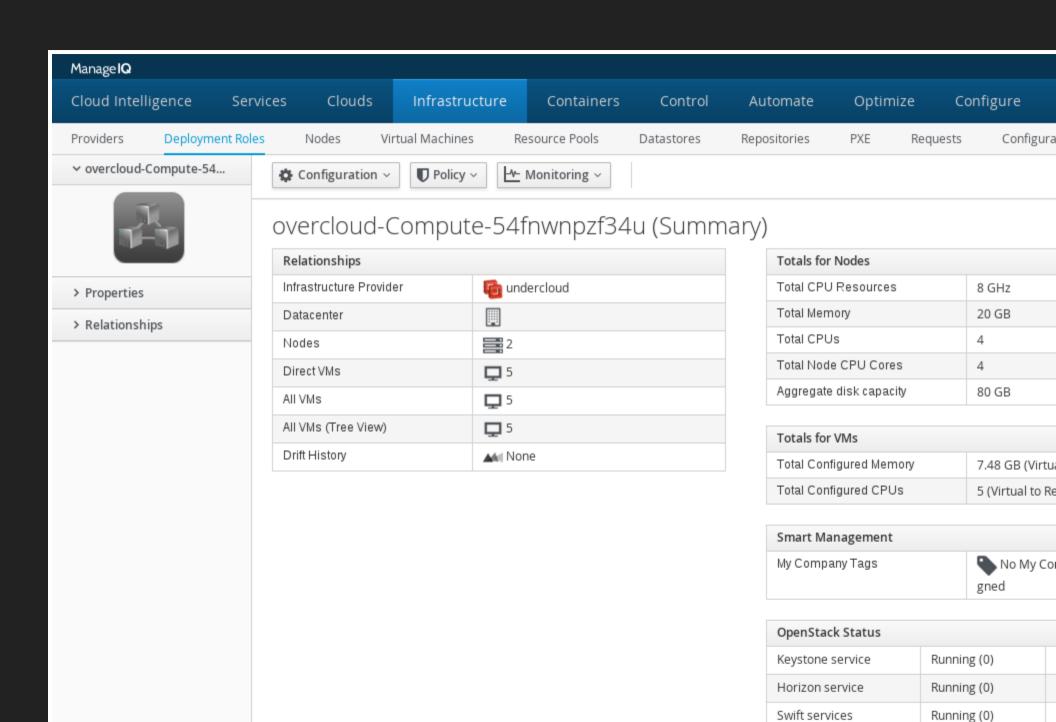
Status	
AMQP Credentials	Valid
Default Credentials	Valid
Last Refresh	Success - About 1 Hour Ago
States of Root Orchestration Stacks	overcloud status: UPDATE_COMPLETE

Relationships	
Deployment Roles	₩ 3
Nodes	5
Cloud Tenants	♣ 1
Availability Zones	♣ 3
Templates	₽ 5
Orchestration stacks	≣ 121
Smart Management	
Managed by Zone	a default
My Company Tags	No My Company Tags have been assig ned

Deployment Role Inventory







Running (2)

Nova services

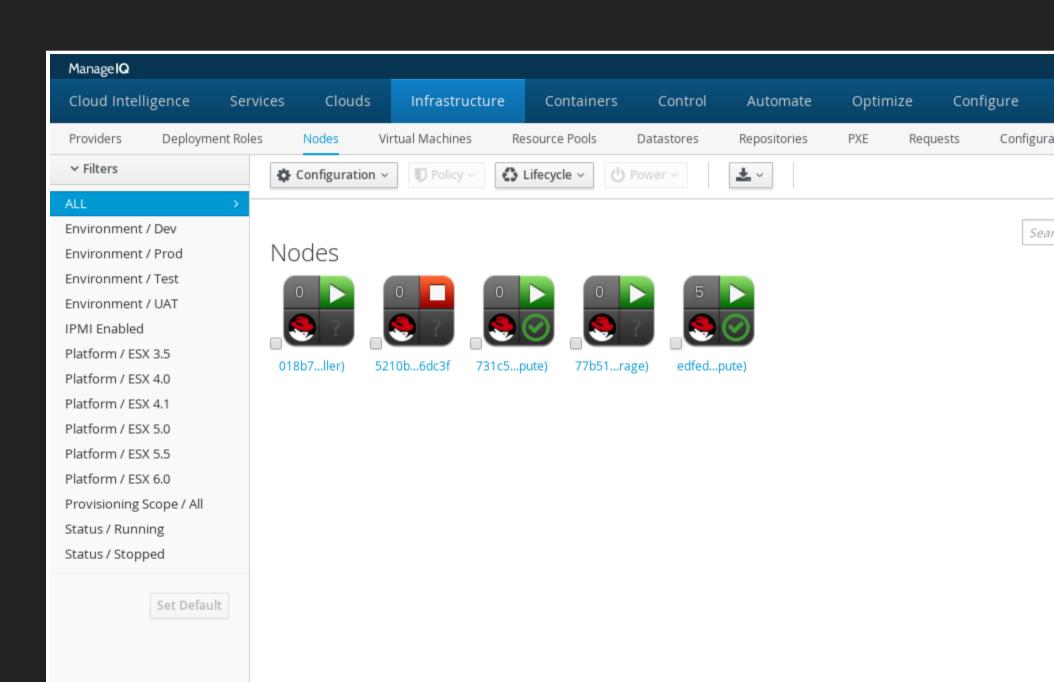
neutron services	Kunning (2)
Ceilometer services	Running (2)
Cinder services	Running (0)
Heat services	Running (0)
Support services	Running (2)
Glance services	Running (0)

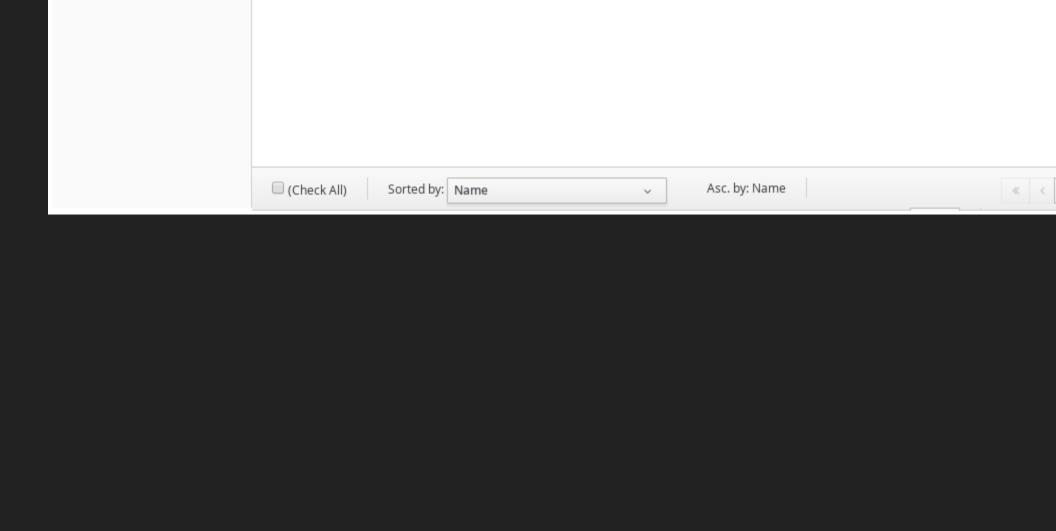
Relationships	
Infrastructure Provider	andercloud undercloud
Datacenter	
Nodes	2
Direct VMs	□ 5
All VMs	□ 5
All VMs (Tree View)	_ 5
Drift History	∆ 6(12

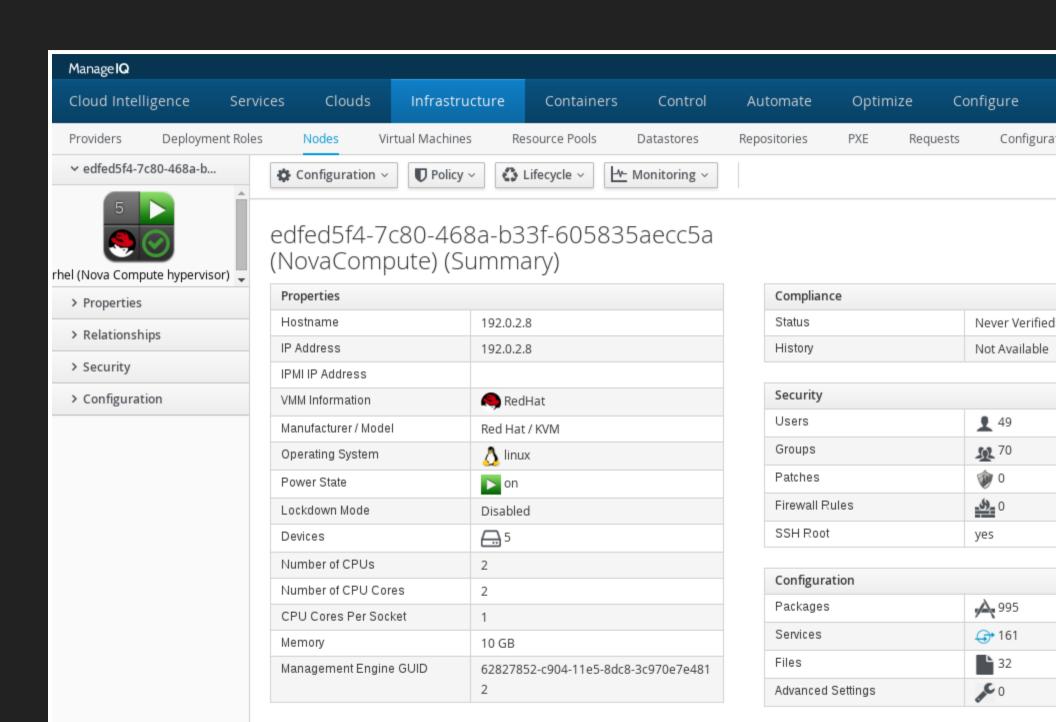
Totals for Nodes	
Total CPU Resources	8 GHz
Total Memory	20 GB
Total CPUs	4
Total Node CPU Cores	4
Aggregate disk capacity	80 GB
Totals for VMs	
Total Configured Memory	7.48 GB (Virtual to Real Ratio: 0.4)
Total Configured CPUs	5 (Virtual to Real Ratio: 1.3)
Smart Management	
My Company Tags	No My Company Tags have been assi gned

OpenStack Status			
Keystone service	Running (0)	Failed (0)	All (0)
Horizon service	Running (0)	Failed (0)	■ All (0)
Swift services	Running (0)	Failed (0)	■ All (0)
Nova services	Running (2)	Failed (0)	■ All (2)
neutron services	Running (2)	Failed (0)	■ All (2)
Ceilometer services	Running (2)	Failed (0)	All (2)
Cinder services	Running (0)	Failed (0)	≅ All (0)
Heat services	Running (0)	Failed (0)	≅ All (0)
Support services	Running (2)	Failed (0)	All (2)
Glance services	Running (0)	Failed (0)	All (0)

Nodes Inventory







m undercloud

Relationships

Infrastructure Provider

Smart Management

My Company Tags

No My Cor

		——————————————————————————————————————
Deployment Role	🖧 overcloud-Compute-54fnwnpzf34u	gned
Availability Zone	a nova	
Cloud Tenants	<u>♣</u> 1	Authentication Status
\/\d-	_	SSH keypair Credentials Valid
VMs	₽ 5	· · · · · · · · · · · · · · · · · · ·
Drift History		OpenStack Status
		Nova services Running Failed C

Properties	
Hostname	192.0.2.8
IP Address	192.0.2.8
IPMI IP Address	
VMM Information	RedHat
Manufacturer / Model	Red Hat / KVM
Operating System	<u></u> linux
Power State	on
Lockdown Mode	Disabled
Devices	<u></u> 5
Number of CPUs	2
Number of CPU Cores	2
CPU Cores Per Socket	1
Memory	10 GB
Management Engine GUID	62827852-c904-11e5-8dc8-3c970e7e481 2

Nodes » edfed5f4-7c80-468a-b33f-605835aecc5a (NovaCompute) (Devices)

edfed5f4-7c80-468a-b33f-605835aecc5a (NovaCompute) (Devices)

Processors

2 (2 sockets x 1 core)

CPU Type

RHEL 7.1.0 PC (i440FX + PIIX, 1996)

CPU Speed

2000 MHz

Memory

\infty 10240 MB

Scsi , Size: 44 GB,

Percent Used

Provisioned Space:

0.0%

Ata-QEMU_HARDDISK_QM00005, Mode: persistent

Relationships	
Infrastructure Provider	andercloud
Deployment Role	🚜 overcloud-Compute-54fnwnpzf34u
Availability Zone	anova 🌊 nova
Cloud Tenants	♣ 1
VMs	₽ 5
Drift History	∆ 6(2

Smart state analysis details in the node

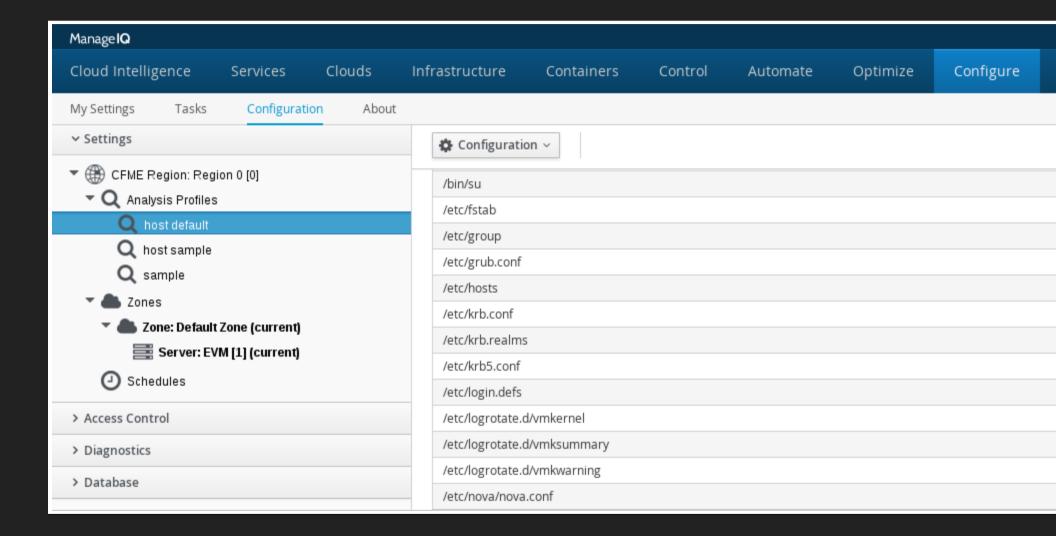
Security	
Users	1 49
Groups	№ 70
Patches	ॐ 0
Firewall Rules	0
SSH Root	yes

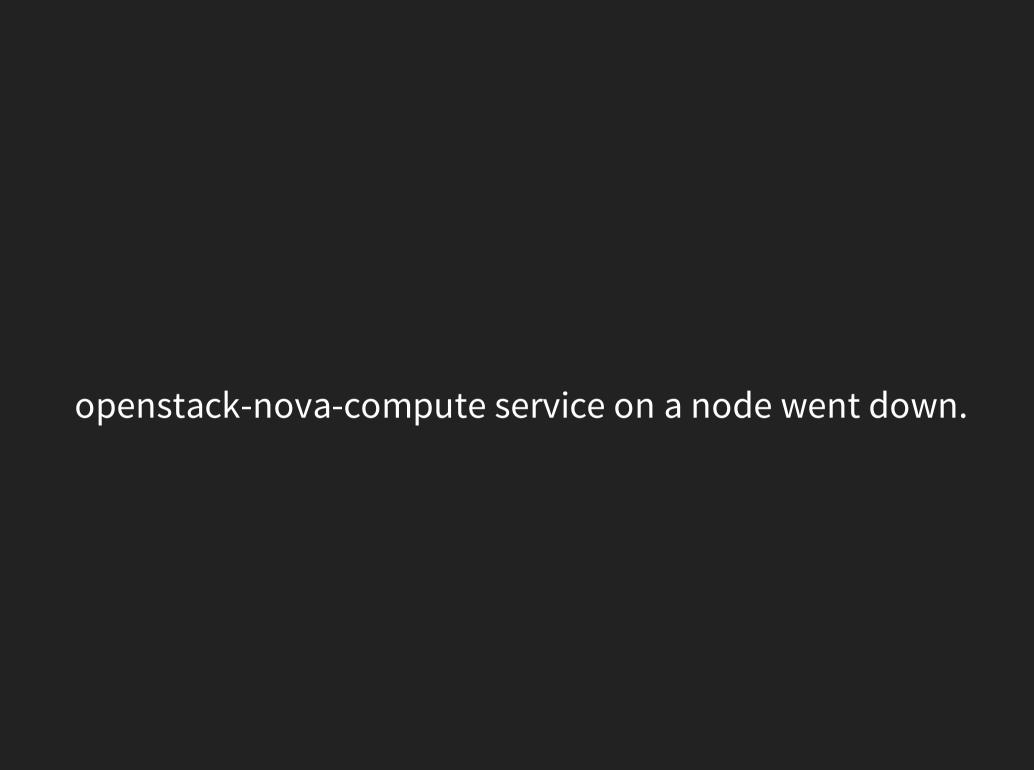
Configuration	
Packages	995
Services	→ 161
Files	33
Advanced Settings	,€ 0

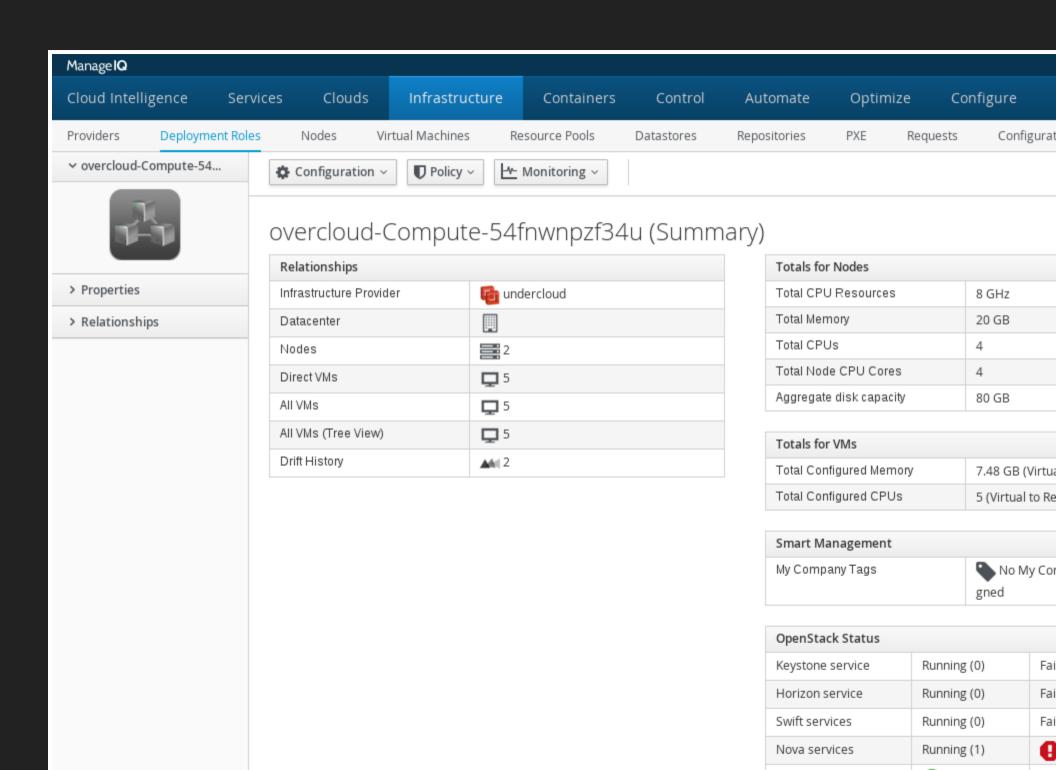
OpenStack Status				
Nova services	Running (1)	Failed (0)	→ All (1)	Configuration (0)
Glance services	Running (0)	Failed (0)	→ All (0)	Configuration (0)
Keystone service	Running (0)	Failed (0)	→ All (0)	Configuration (0)
Horizon service	Running (0)	Failed (0)	→ All (0)	Configuration (0)
neutron services	Running (1)	Failed (0)	→ All (3)	Configuration (0)
Swift services	Running (0)	Failed (0)	→ All (0)	Configuration (0)
Cinder services	Running (0)	Failed (0)	→ All (0)	Configuration (0)
Ceilometer services	Running (1)	Failed (0)	→ All (1)	Configuration (0)
Heat services	Running (0)	Failed (0)	→ All (0)	Configuration (0)
Support services	Running (2)	Failed (0)	→ All (3)	Configuration (0)

SMART STATE

Specify what files to collect by smartstate

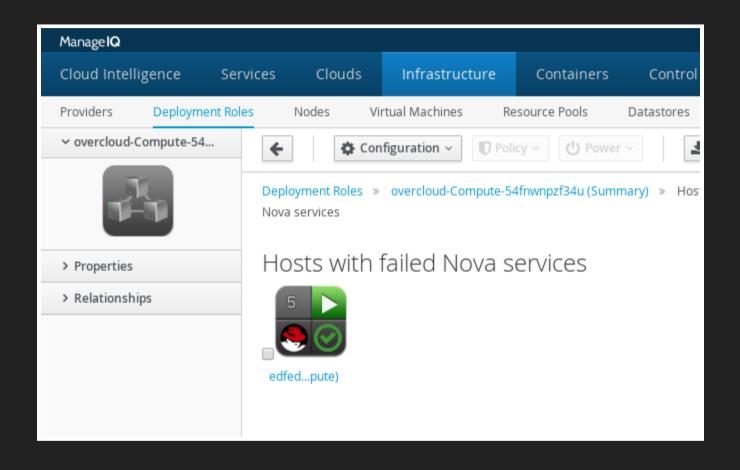




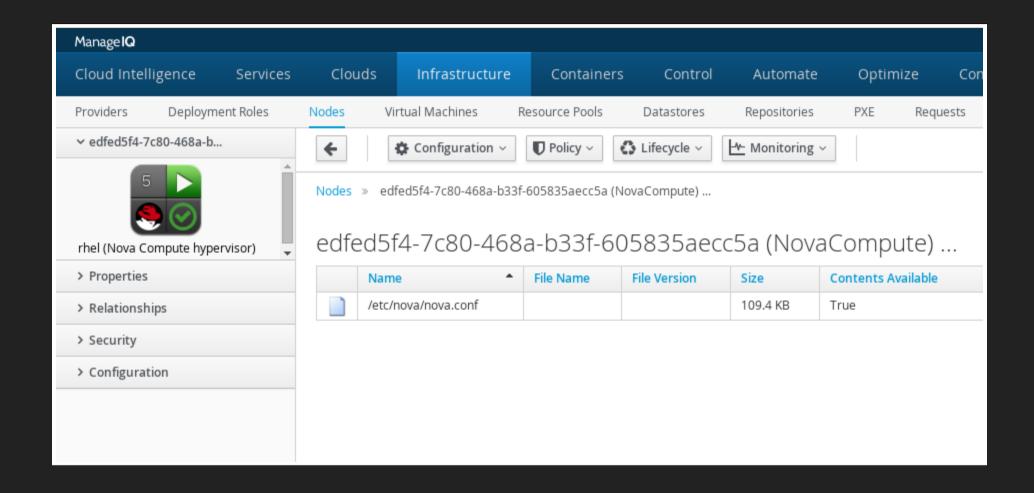


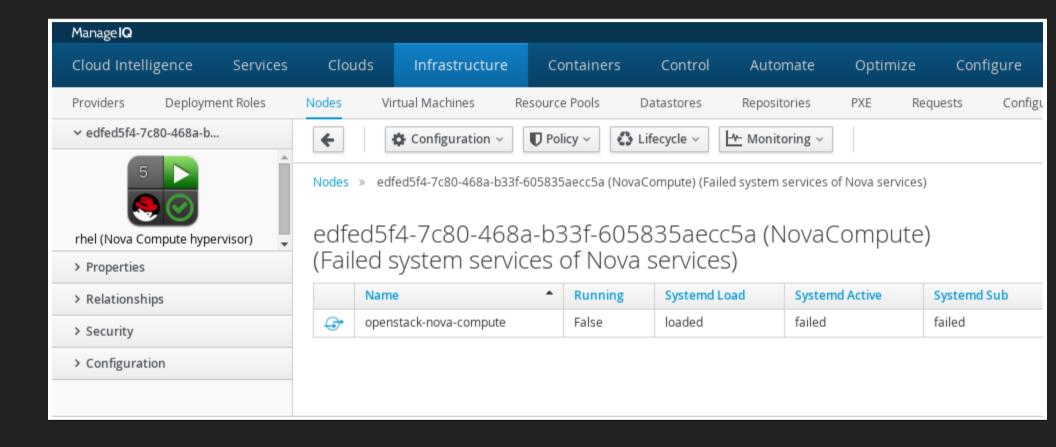
Fai	Running (2)	neutron services
Fai	Running (2)	Ceilometer services
Fai	Running (0)	Cinder services
Fai	Running (0)	Heat services
Fai	Running (2)	Support services
Fai	Running (0)	Glance services

OpenStack Status			
Keystone service	Running (0)	Failed (0)	All (0)
Horizon service	Running (0)	Failed (0)	All (0)
Swift services	Running (0)	Failed (0)	≅ All (0)
Nova services	Running (1)	Failed (1)	All (2)
neutron services	Running (2)	Failed (0)	≅ All (2)
Ceilometer services	Running (2)	Failed (0)	All (2)
Cinder services	Running (0)	Failed (0)	≅ All (0)
Heat services	Running (0)	Failed (0)	All (0)
Support services	Running (2)	Failed (0)	All (2)
Glance services	Running (0)	Failed (0)	All (0)

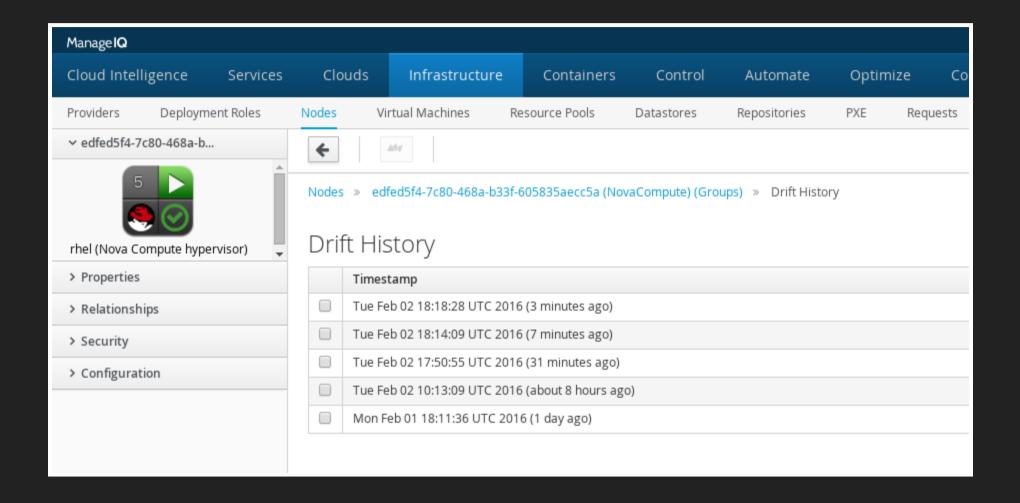


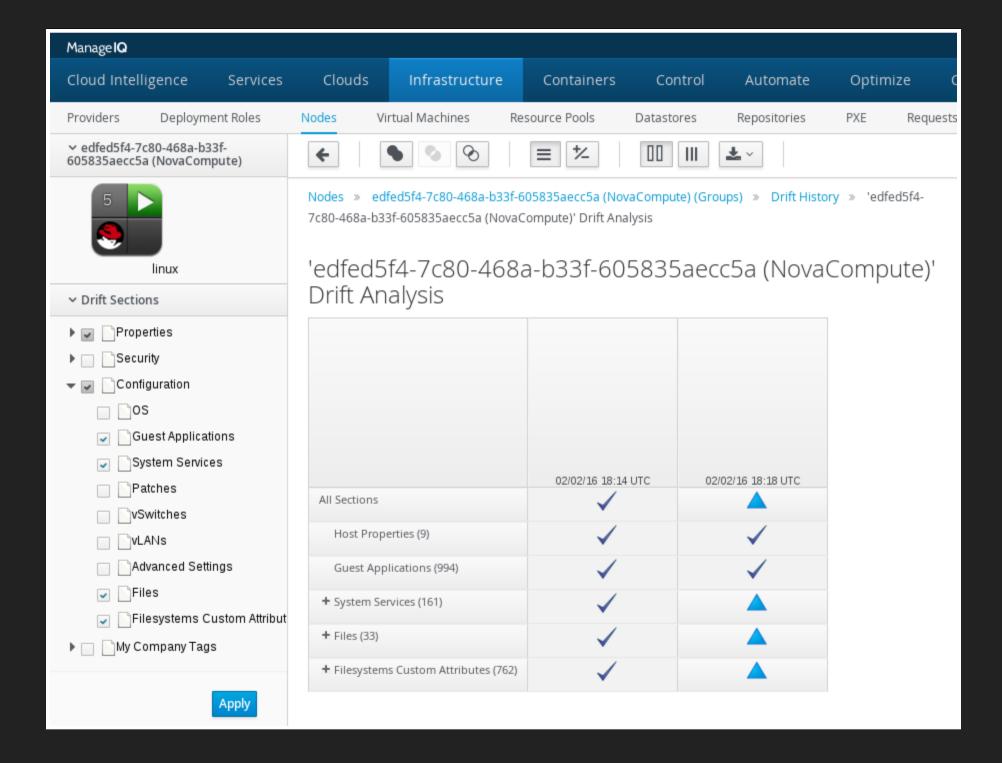
OpenStack Status				
Nova services	Running (0)	Failed (1)	All (1)	Configuration (1)
Glance services	Running (0)	Failed (0)	→ All (0)	Configuration (0)
Keystone service	Running (0)	Failed (0)	→ All (0)	Configuration (0)
Horizon service	Running (0)	Failed (0)	→ All (0)	Configuration (0)
neutron services	Running (1)	Failed (0)	→ All (3)	Configuration (0)
Swift services	Running (0)	Failed (0)	→ All (0)	Configuration (0)
Cinder services	Running (0)	Failed (0)	→ All (0)	Configuration (0)
Ceilometer services	Running (1)	Failed (0)	→ All (1)	Configuration (0)
Heat services	Running (0)	Failed (0)	→ All (0)	Configuration (0)
Support services	Running (2)	Failed (0)	→ All (3)	Configuration (0)





DRIFT STATE



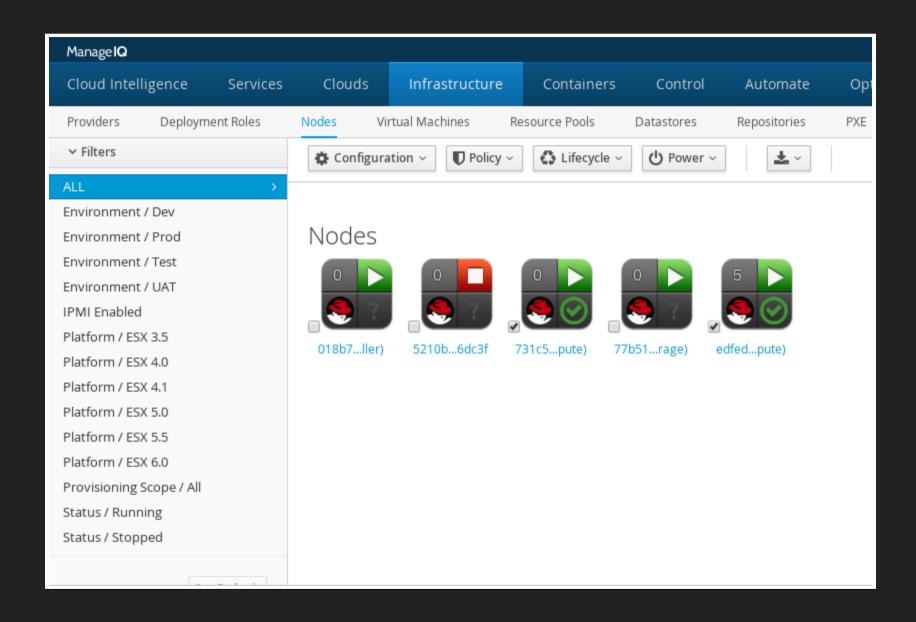


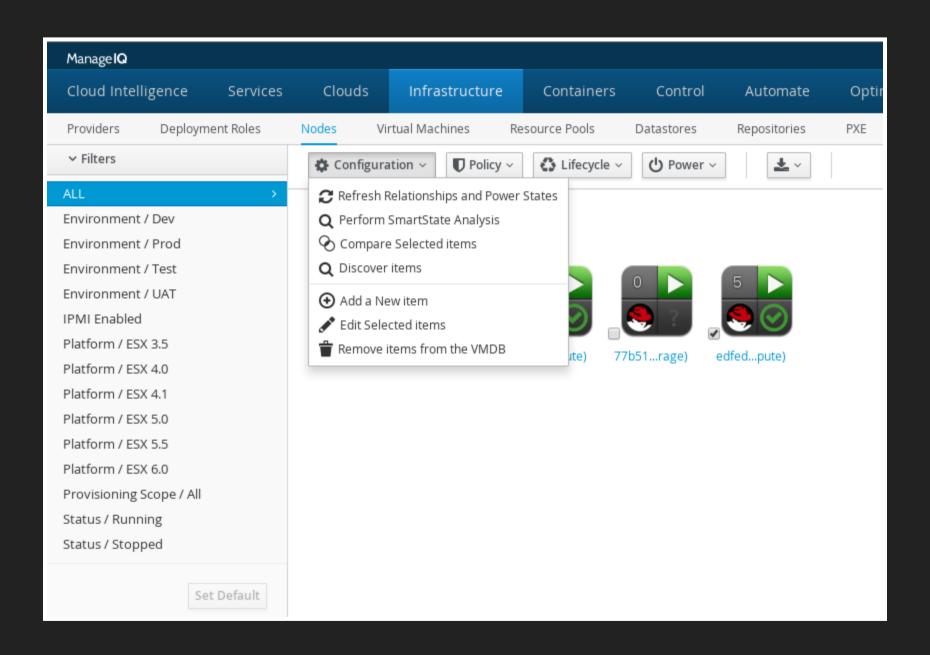
System Services (161)	✓	A
openstack-nova-compute	✓	A
Display Name	(empty)	(empty)
Running	true	false

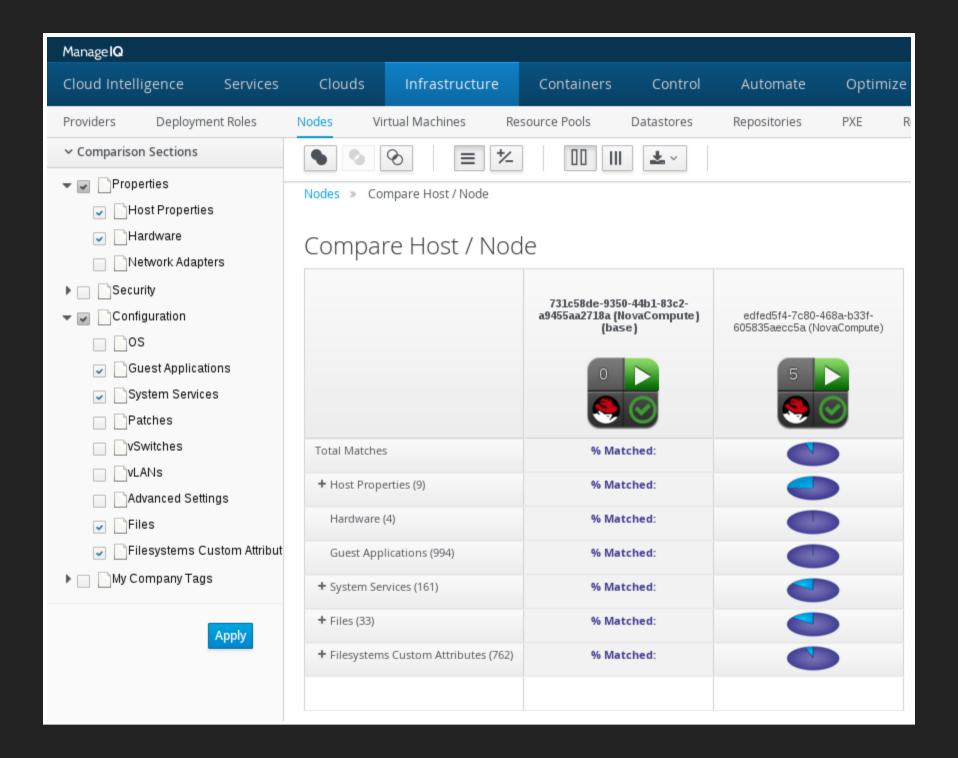
= Files (33)	✓	_
<pre>/etc/nova/nova.conf</pre>	✓	A
md5	ea0c34f4a4c2368e8494b70	23e8a2894859bcd06d9eea
Size	111997	111996
Permissions	0640	0640
Owner	nova	nova
Group	nova	nova
Mtime	Tue Feb 02 18:07:44 UTC 20	Tue Feb 02 18:16:33 UTC 20
± /var/log/messages	✓	A
± /var/log/secure	✓	A

Filesystems Custom Attributes (762)	✓	A
= /etc/nova/nova.conf:DEFAULT:r	✓	A
Name	rpc_backend	rpc_backend
Section	DEFAULT	DEFAULT
Value	rabbit	bunny
Value Interpolated	rabbit	bunny
Source	defined	defined

COMPARING NODES







Host properties diff

− Host Properties (9)	% Matched:	
Name	731c58de-9350-44b1-83c2- a9455aa2718a (NovaCompute)	edfed5f4-7c80-468a-b33f- 605835aecc5a (NovaCompute)
Number Of VMs	0	5
Hardware (4)	% Matched:	
Guest Applications (994)	% Matched:	

Host services diff

dhcp-interface@tapf3832e82-24	% Matched:	
Display Name	(missing)	(empty)
Running	(missing)	false
■ openstack-nova-compute	% Matched:	
Display Name	(empty)	(empty)
Running	true	false

Host files diff

= /etc/hosts	% Matched:	
md5	87eb3b7efd8e18ed63012face9b9d	87eb3b7efd8e18ed63012face9b9c9
Size	526	526 bytes
Permissions	0644	0644
Owner	root	root
Group	root	root
Mtime	Mon Feb 01 19:06:15 UTC 2016	Mon Feb 01 19:04:46 UTC 2016

= /etc/nova/nova.conf	% Matched:	
md5	ef3736bca6f0bd3c4e202895b05b7	23e8a2894859bcd06d9eeae1f39bcc
Size	111998	111,996 bytes
Permissions	0640	0640
Owner	nova	nova
Group	nova	nova
Mtime	Mon Feb 01 19:05:09 UTC 2016	Tue Feb 02 18:16:33 UTC 2016

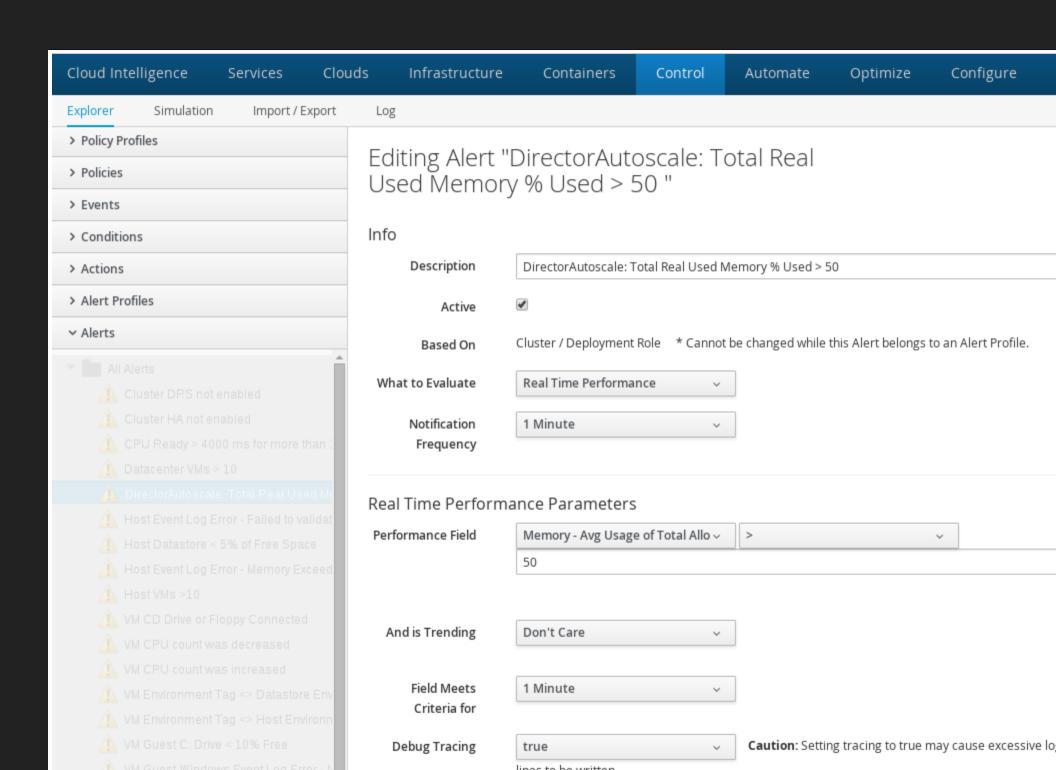
Host custom attributes diff

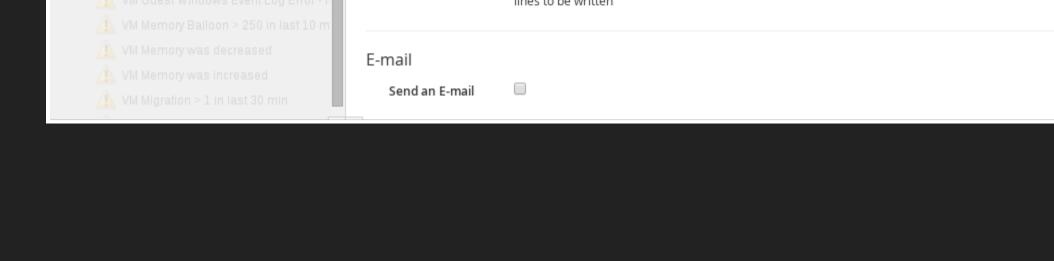
= /etc/nova/nova.conf:DEFAULT:r	% Matched:	
Name	rpc_backend	rpc_backend
Section	DEFAULT	DEFAULT
Value	rabbit	bunny
Value Interpolated	rabbit	bunny
Source	defined	defined

= /etc/nova/nova.conf:DEFAULT:	% Matched:	
Name	my_block_storage_ip	my_block_storage_ip
Section	DEFAULT	DEFAULT
Value	\$my_ip	\$my_ip
Value Interpolated	192.0.2.10	192.0.2.8
Source	default	default

AUTOSCALING OF COMPUTE HOSTS USING AUTOMATE

Define an alert





Management Event

Send a

1

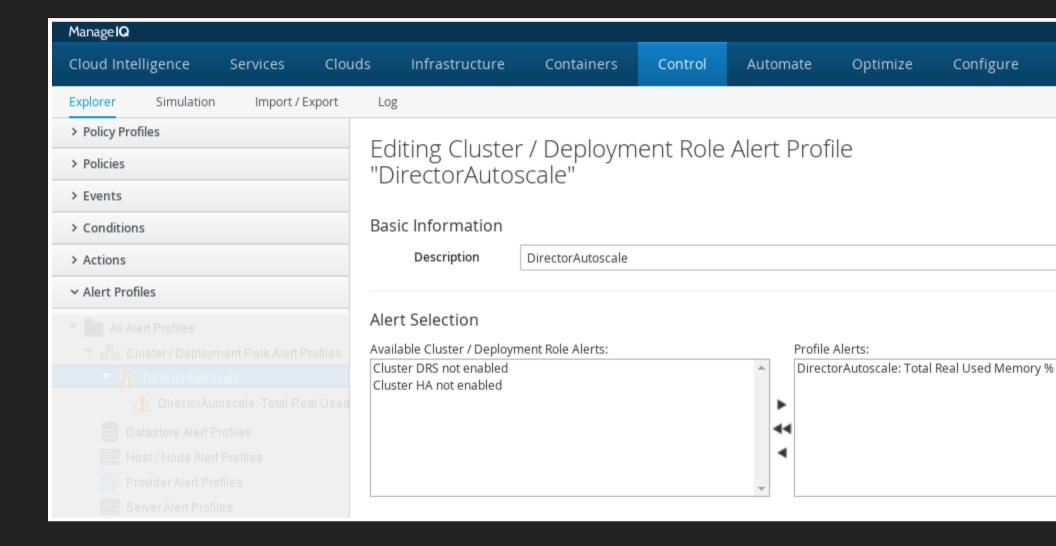
Management

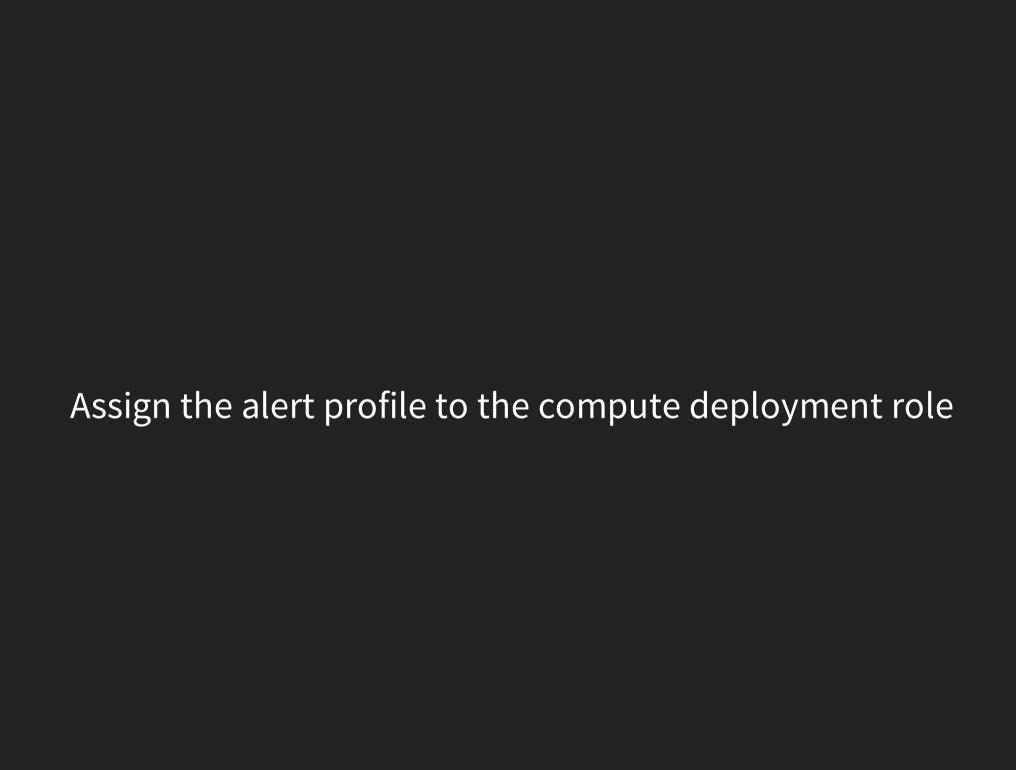
Event

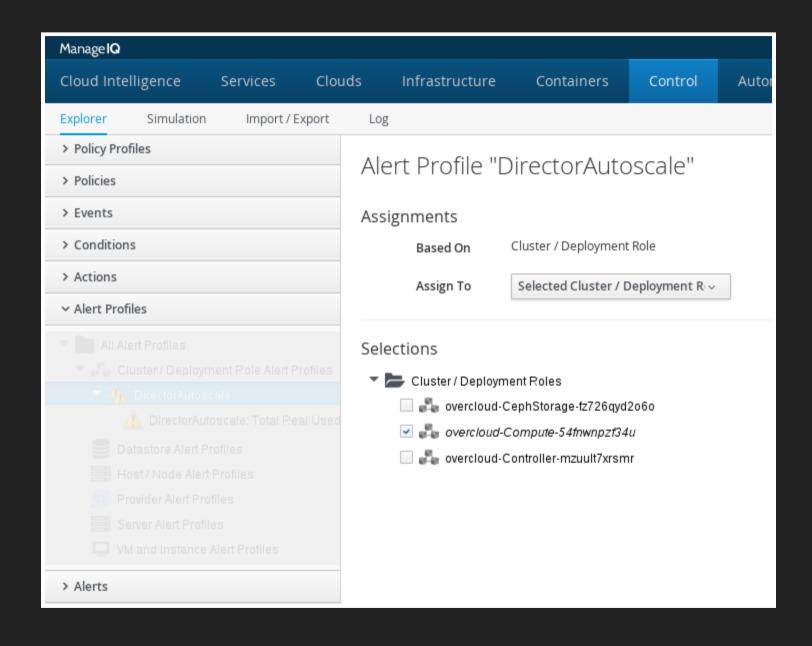
Event Name

DirectorAutoscale_Processing

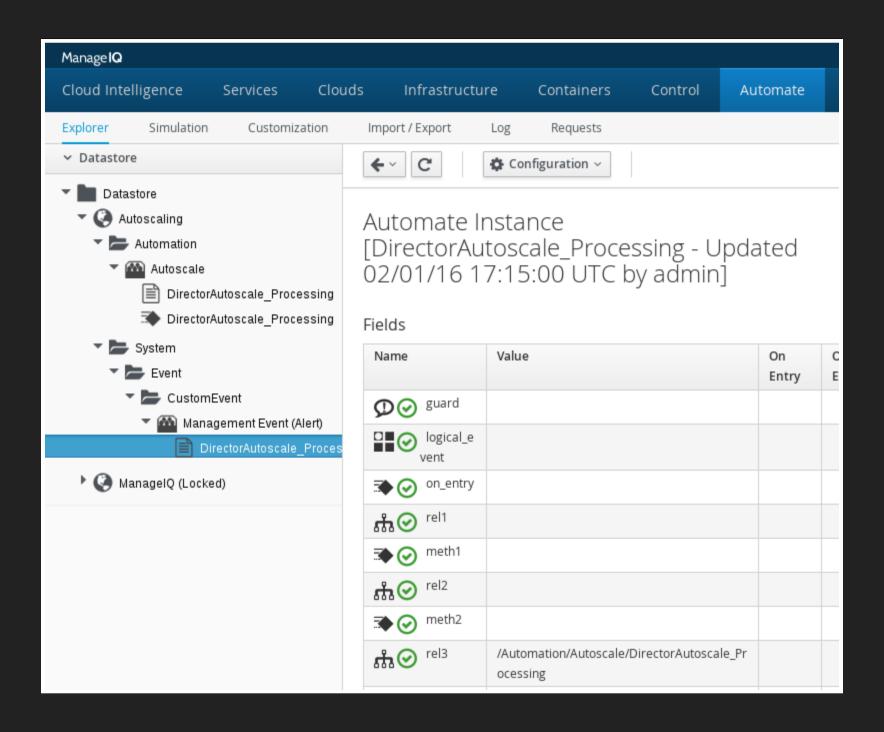
Create alert profile

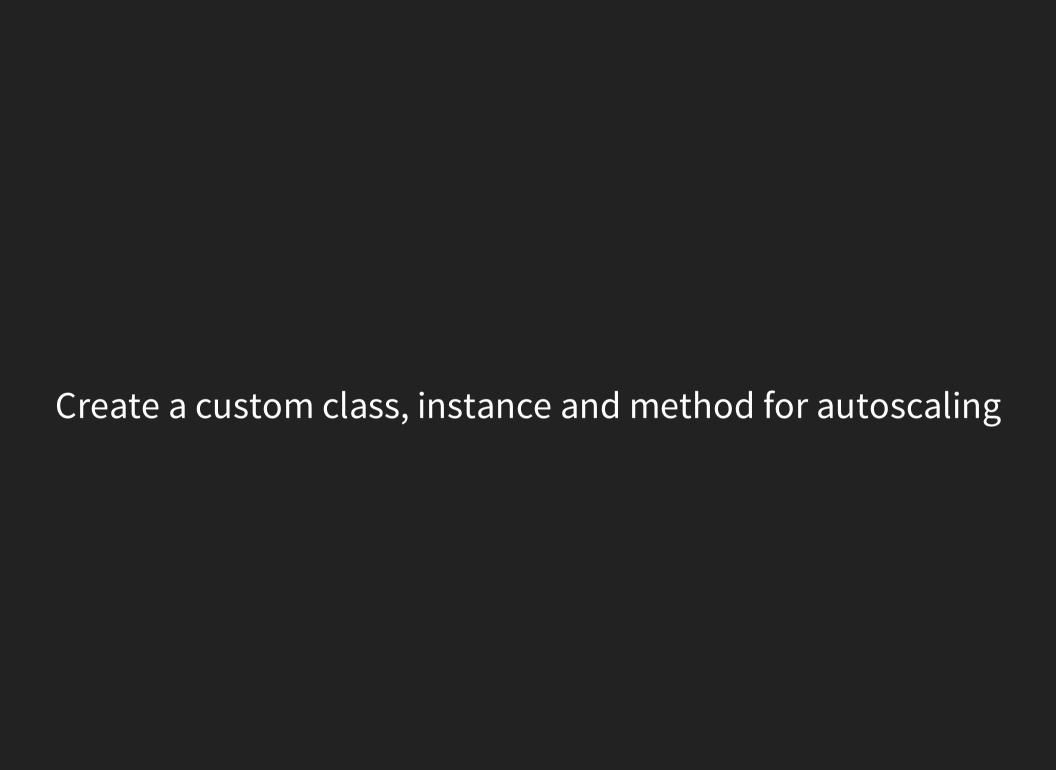


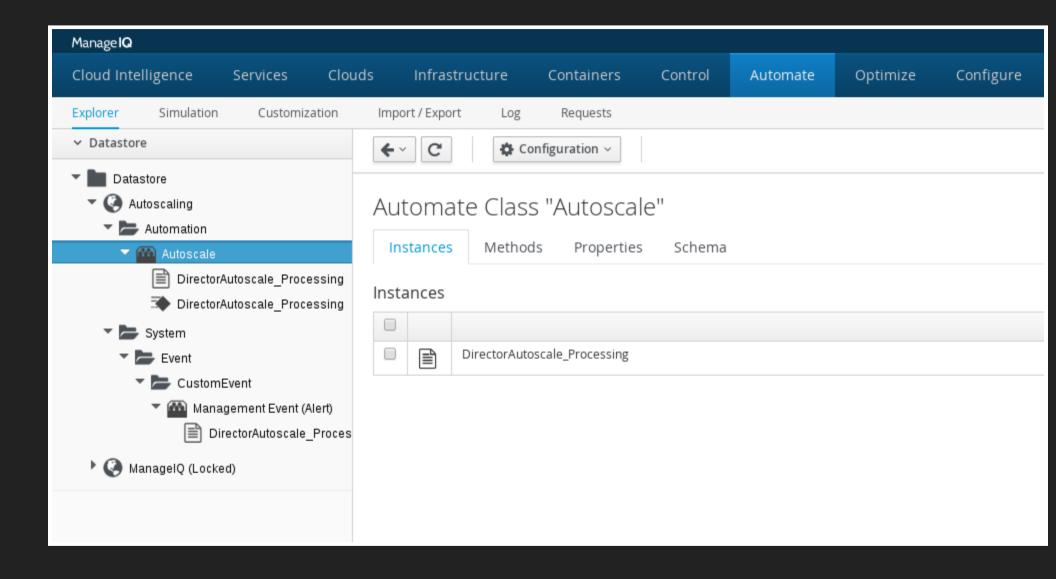


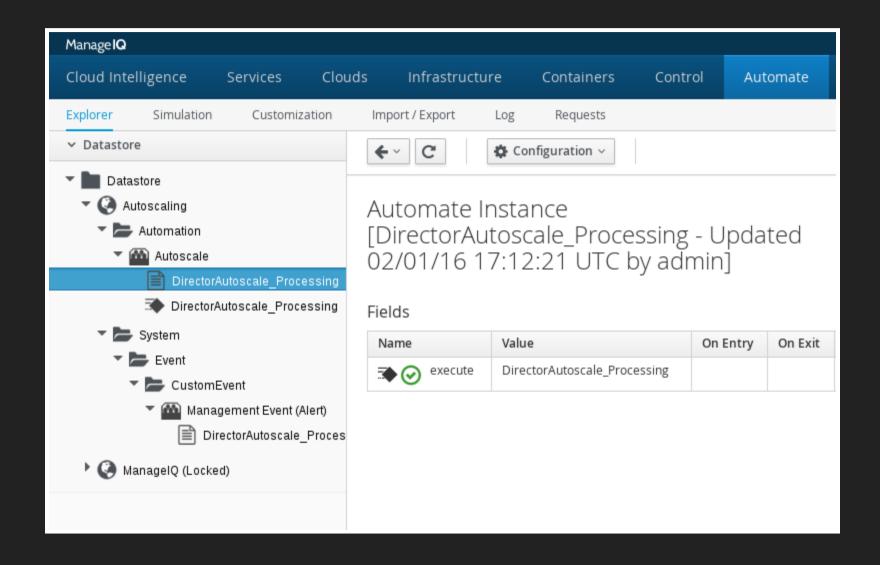


Create a custom event handler



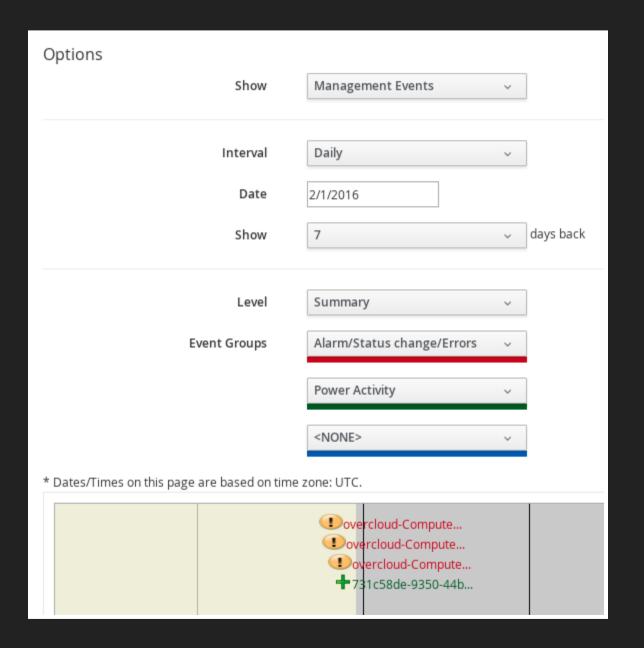


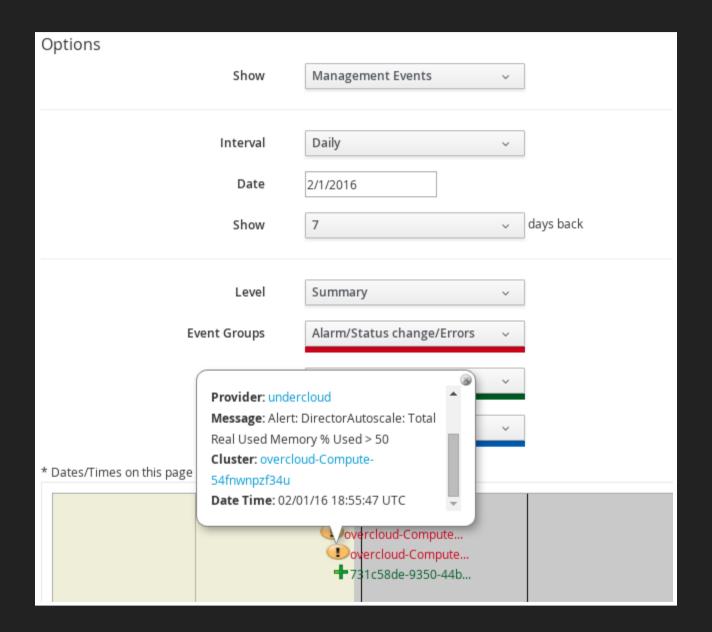


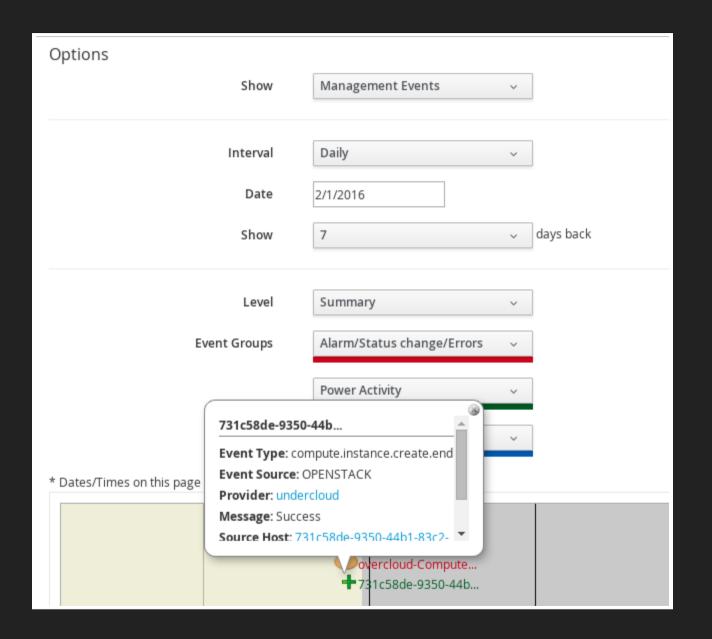


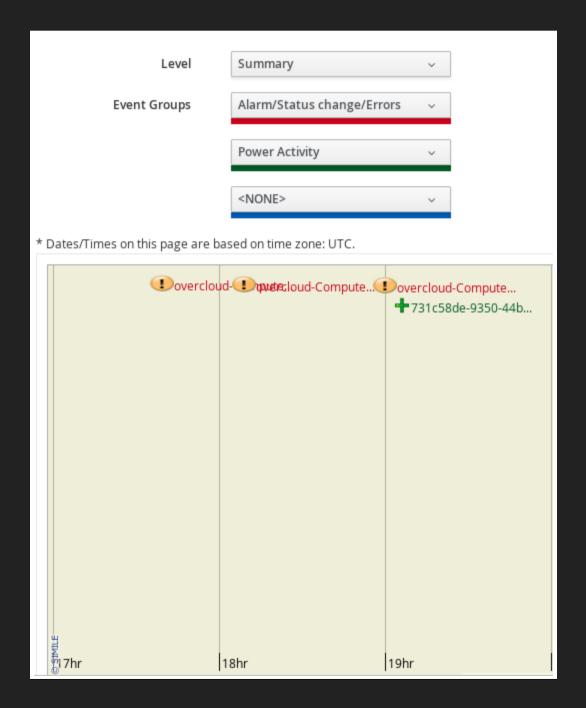
```
13 begin
14 # Method for logging
15
     def log(level, message)
16
17
       @method = 'InstackAutoscale Processing'
       $evm.log(level, "#{@method}: #{message}")
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
     log(:info, "CFME Autoscaling Method Started")
     # Get cluster(deployment role)
     cluster = $evm.root['ems_cluster']
     raise "Cluster object not found" if cluster.nil?
     log(:info, "Inspecting cluster object: <#{cluster.inspect}>")
     # Get Ems
     ems = cluster.ext management system
     # Get current number of compute hosts
     hosts count = cluster.hosts.count
     # Get orchestration stack
     stack = ems.direct orchestration stacks.first
     # Scale up by one host if free hosts are available, Nodes that are powered off are free in this example
     if ems.hosts.select{ |x| x.power_state == 'off' }.count > 0
       # There are free nodes, lets scale +1
38
39
       hosts count += 1
40
       log(:Info, "Autoscaling to #{hosts count}")
41
       stack.raw_update_stack(nil, {"ComputeCount" => hosts_count.to_s })
42
43
       log(:warn, "Not enough free hosts for autoscale")
44
     end
45
46 end
47
```

Observe an autoscaling in the timeline



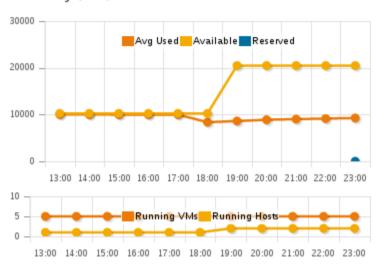




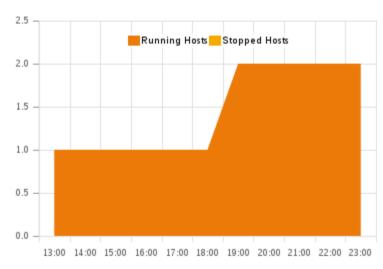


Observe an autoscaling in the charts

Memory (MB)



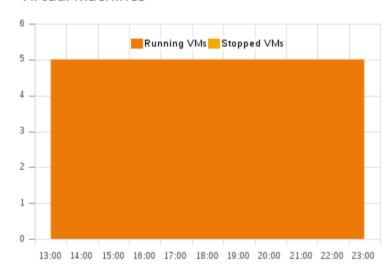
Hosts



Disk I/O (blocks per second)



Virtual Machines



Q&A

Contacts:

- My GitHub https://github.com/Ladas
- Ask me/us directly on: https://gitter.im/ManageIQ/manageiq
- ManagelQ main repo https://github.com/ManagelQ/manageiq
- Yes, we are hiring!