

# Import/Export OVA

Arik Hadas  
Deep Dive

- Existing OVA support:
  - Importing VMware-compatible OVAs (external OVAs)
- Desired OVA support:
  - Exporting oVirt-VMs to OVA
    - An alternative to export domains\*
  - Exporting oVirt-templates to OVA
    - An alternative to export domains
    - To share templates between RHV environments
  - Importing oVirt-OVAs

# oVirt Exporting oVirt-VMs to OVAs

New Edit Remove Run Suspend Shutdown Reboot Console Migrate Create Snapshot

enter	Memory	CPU	Network	Graphics	Status	Uptime	Description
t	--	--	--	None	Down		
t	--	--	--	None	Down		

- Import
- Clone VM
- Change CD
- Cancel Migration
- Cancel Conversion
- Make Template
- Export to Export Domain
- Export as OVA
- Assign Tags
- Guide Me

# oVirt Exporting oVirt-VMs to OVAs

Export Virtual Machine(s) As Virtual Appliance

Host

ziggy

Directory

Name

fedora.ova

OK

Cancel

# oVirt Exporting oVirt-VMs to OVAs

```
# Create the connection to the server:
connection = sdk.Connection(
    url='https://engine40.example.com/ovirt-engine/api',
    username='admin@internal',
    password='redhat123',
    ca_file='/etc/pki/ovirt-engine/ca.pem',
    debug=True,
    log=logging.getLogger(),
)

# Find the virtual machine:
vms_service = connection.system_service().vms_service()
vm = vms_service.list(search='name=myvm')[0]
vm_service = vms_service.vm_service(vm.id)

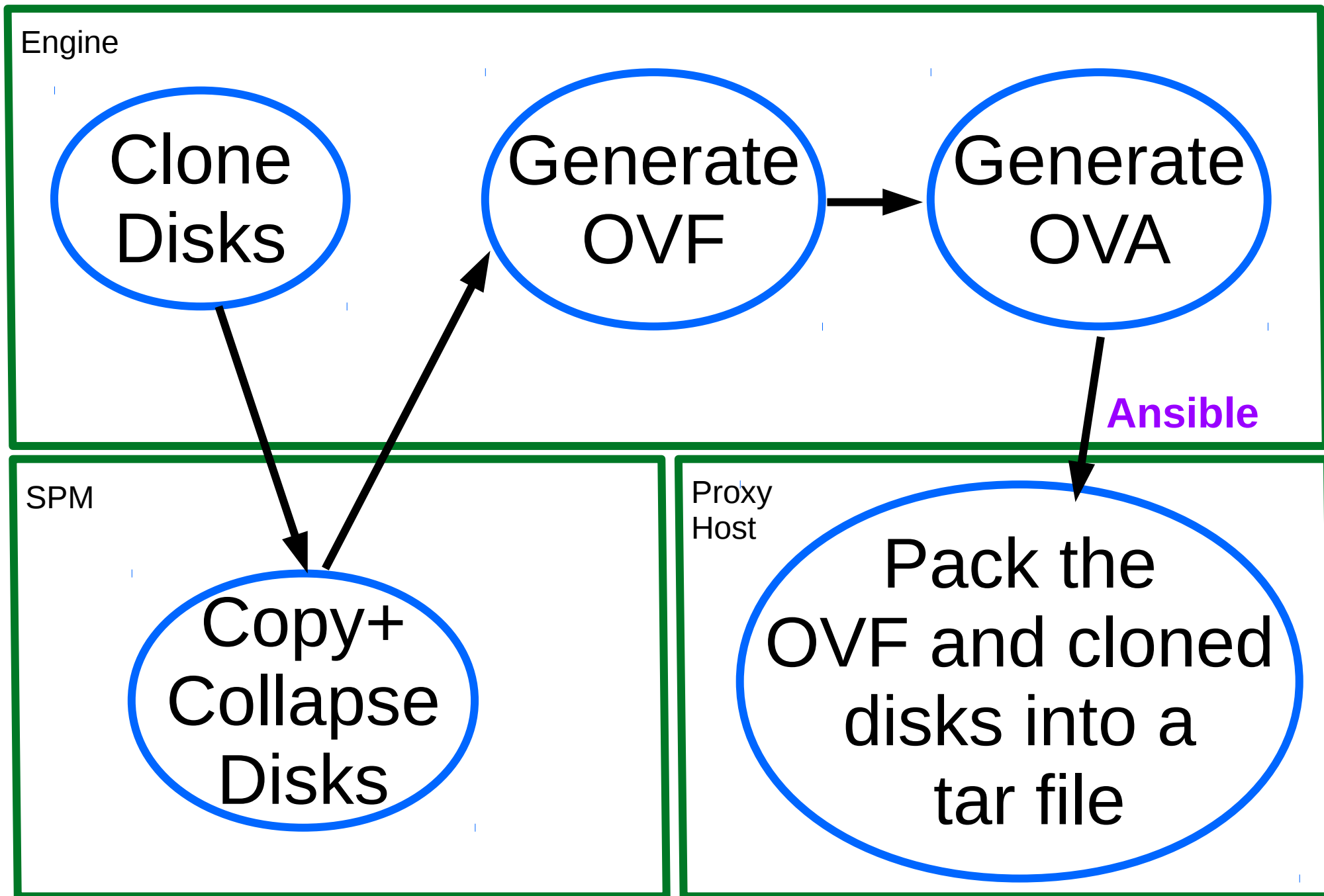
# Find the host:
hosts_service = connection.system_service().hosts_service()
host = hosts_service.list(search='name=myhost')[0]

# Export the virtual machine. Note that the 'filename' parameter is
# optional, and only required if you want to specify a name for the
# generated OVA file that is different from <vm_name>.ova.
# Note that this operation is only available since version 4.2 of
# the engine and since version 4.2 of the SDK.
vm_service.export_to_path_on_host(
    host=types.Host(id=host.id),
    directory='/tmp',
    filename='myvm2.ova'
)

# Close the connection to the server:
connection.close()
```

[Full SDK Example \(github\)](#)

# oVirt Exporting oVirt-VMs to OVAs



# oVirt Exporting oVirt-VMs to OVAs

engine.log

```
CopyAllOvaDisksCommand  
CopyImageGroupCommand  
....  
CopyImageGroupCommand
```

```
CreateOvaCommand
```

ovirt-export-ova...log

```
Playbook ovirt - ova - export
```

engine.log

```
RemoveDiskCommand  
....  
RemoveDiskCommand
```

# Playbook ovirt-ova-export

Role: ovirt-ova-export-pre-pack

Validate the target folder

Prepare <ova name>.ova.tmp file

Role: ovirt-ova-pack

Pack OVF and disk into the file

Role: ovirt-ova-export-post-pack

Rename to <ova\_name>.ova

Remove OVA file on failure



# Exporting oVirt-templates to OVAs

- Not available yet
  - Planned for the next release
- Very similar process to export VM to OVA
- Provides a way of sharing templates

# Importing oVirt-OVAs

Import Virtual Machine(s)

X

Data Center

Default

▼

Source

Export Domain

▼

Name

VMware

Export Domain

Virtual Appliance (OVA)

XEN (via RHEL)

KVM (via Libvirt)

Path

Description

Load

Virtual Machines on Source

☐

Name

Virtual Machines to Import

☐

Name

➡

⬅

Next

Cancel

# Importing oVirt-OVAs

## Import Virtual Machine(s)



Data Center

Default



Source

Virtual Appliance (OV



Host

ziggy



File Path

/tmp/my\_fedora.ova

Load

### Virtual Machines on Source

<input type="checkbox"/>	Name
<input type="checkbox"/>	my_fedora



### Virtual Machines to Import

<input type="checkbox"/>	Name

Next

Cancel

# Importing oVirt-OVAs

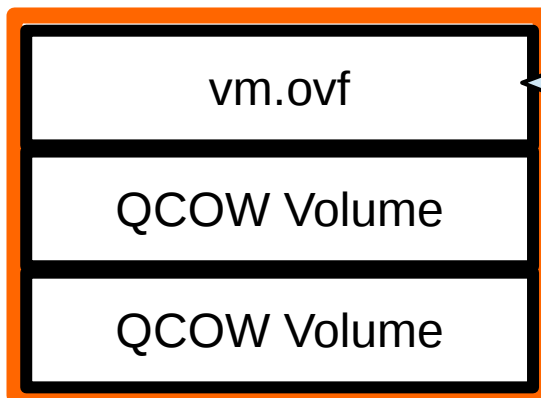
```
imports_service.add(  
    types.ExternalVmImport(  
        name='myvm',  
        provider=types.ExternalVmProviderType.KVM,  
        url='ova:///myvms/my.ova',  
        cluster=types.Cluster(  
            name='mycluster'  
        ),  
        storage_domain=types.StorageDomain(  
            name='mydata'  
        ),  
        host=types.Host(  
            name='myhost'  
        ),  
        drivers_iso=types.File(  
            id='virtio-win.iso'  
        ),  
        sparse=True  
    )  
)
```

[Full SDK Example \(github\)](#)

# Importing oVirt-OVAs

- Similar to importing an external OVA
  - From the client side
- Does not involve virt-v2v
  - Distinguish oVirt-OVAs by 'ovirt' namespace
  - Use the configuration within the OVF
  - Copy the disk volumes to storage domains
- Does not support templates yet

Tar file



```
<ovf:Envelope  
  xmlns:ovf="http://schemas.dmtf.org/ovf/envelope/1"  
  ...  
  xmlns:ovirt="http://www.ovirt.org/ovf">  
  ...
```

# Importing oVirt-OVAs

engine.log

ImportVmFromOvaCommand

AddDiskCommand

....

AddDiskCommand

ExtractOvaCommand

ovirt-import-ova...log

Playbook ovirt-ova-extract

engine.log

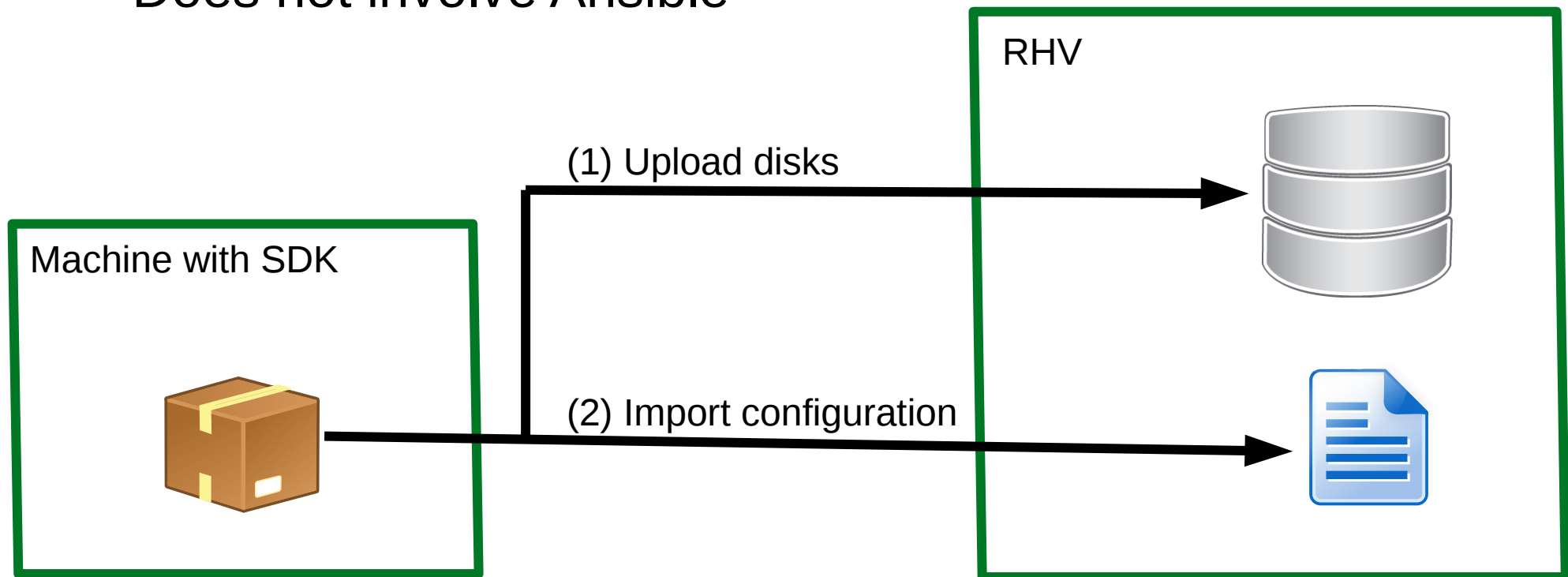
TeardownImageCommand

....

TeardownImageCommand

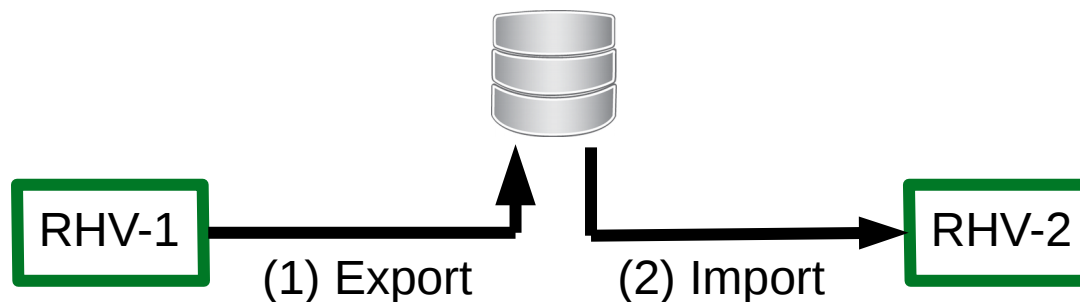
# Uploading OVA

- Create a VM from an OVA that resides out of the DC
  - Currently, limited to VMs
  - In 4.2, limited to oVirt-OVA
  - In 4.2, limited to the SDK
- Does not involve Ansible

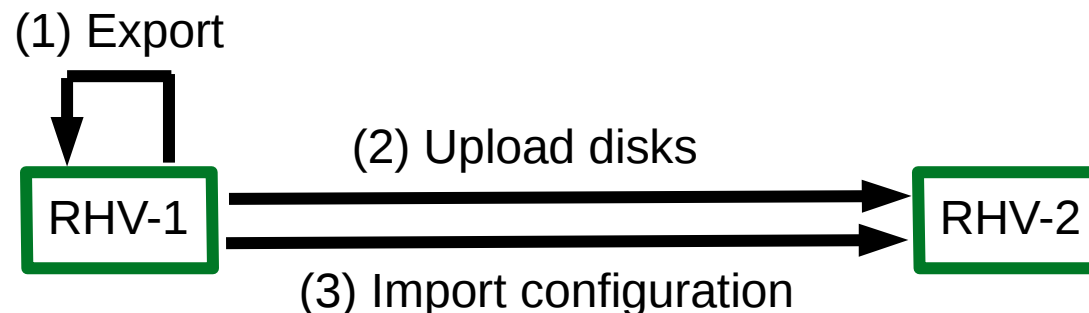


# Sharing Vms/templates Efficiently

- Using a shared storage (TBD: Glance)



- Uploading to another environment





- May introduce new issues with external OVAs
  - Due to unifying OVF parsing
- Not an exact alternative to export domain
  - For instance, OVAs do not contain snapshots
- OVA cannot be produced on block storage
- No support for oVirt-OVA as a folder (rather than tar file)
- ‘File Path’ should point to an OVA
  - Although it may seem that the import dialog supports specifying a folder of OVAs as well
- Using direct I/O

**THANK YOU!**

**Question?**