Configuring backup storage in Ovirt

25.03.2017

Name and Contact Info

Shubham Dubey

Email: sdubey504@gmail.com

Telephone number: +91-9413402251

Postal address: F-192 Ram Nagar Ext, Swej Farm,Sodala,Jaipur

State and country: Rajasthan,India

Links

IRC Handle : shubham0d

GitHub UserName: https://github.com/shubham0d

LinkedIn: https://www.linkedin.com/in/shubham0d/

Resume: https://goo.gl/ic17JS

University info

University: The LNM Institute of Information Technology

Major: Computer Science

Degree: B.tech

Project Idea and Mentor

Idea: Configuring backup storage in Ovirt to support all storage domain.

Mentor:Maor Lipchuk

Synopsis

Ovirt has support for multiple storage domains (a group of storage array)

File Storage Domains

- NFS
- Gluster
- POSIX-Compliant FS
- Local

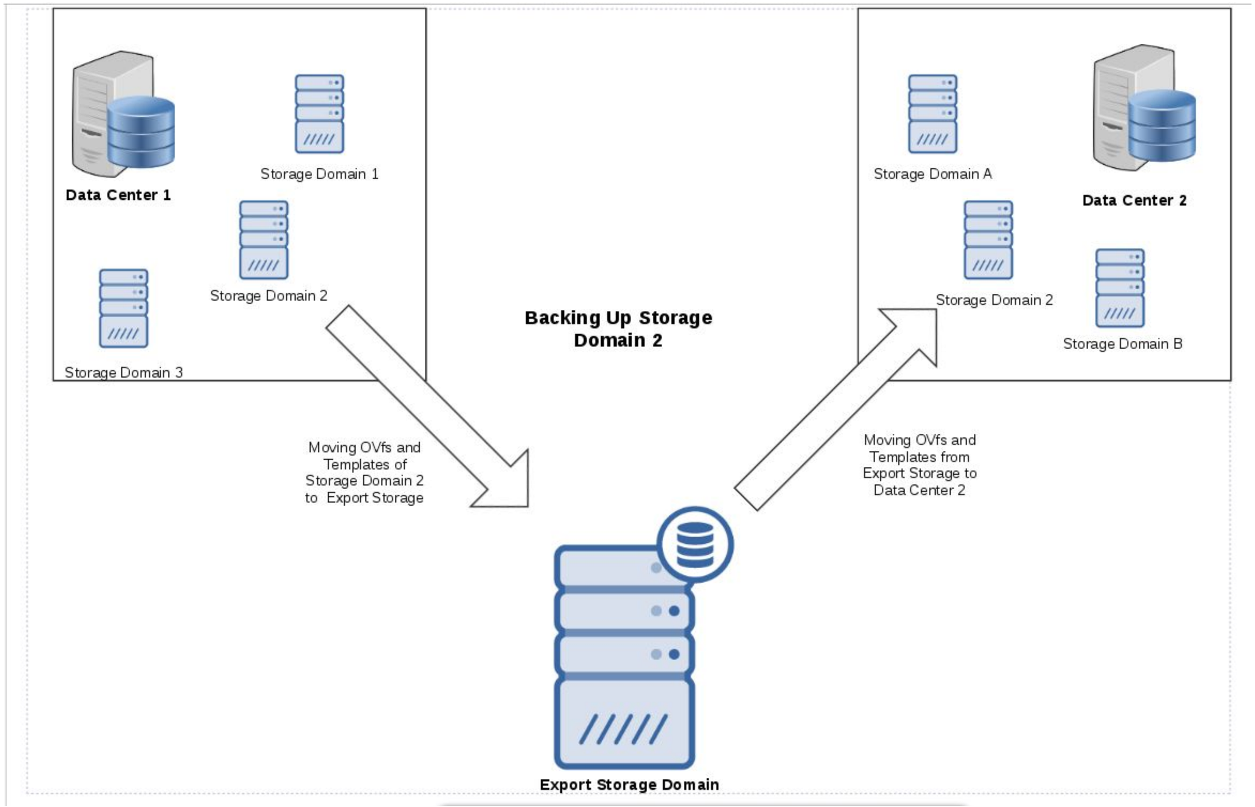
Block Storage Domains

- Fibre Channel
- iSCSI
- Ceph

The project is to add the functionality to set any of supported Storage domain to be used as backup storage in replacement of dedicated backup storage domain(export storage domain).The new backup storage can already be part of any other data center and will be later tagged as backup storage.

Motivation and Benefits to community

Ovirt allows centralized management of virtual machines, compute, storage and networking resources. It let you manage a pool of resources so you can distribute your virtual workloads across them in an efficient manner.

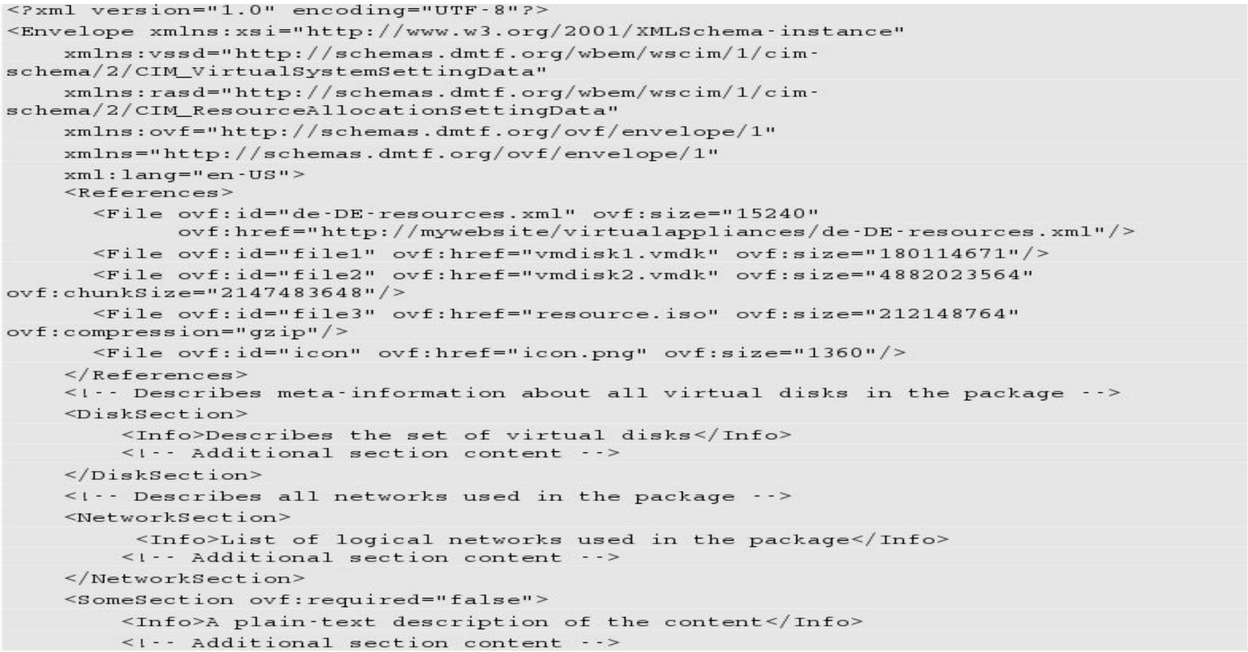

Data Center 1
Storage Domain 1
Storage Domain 2
Storage Domain 3
Backing Up Storage Domain 2
Storage Domain A
Data Center 2
Storage Domain 2
Storage Domain B
Moving OVfs and Templates of Storage Domain 2 to Export Storage
Moving OVfs and Templates from Export Storage to Data Center 2
Export Storage Domain

Ovirt uses export storage domain to the storage backup and disaster recovery.

●Export storage domain is a storage domain specifically used for backup vms and disks.

●It uses OVF (Open virtualization Format) to backup entities.

Fig 1-An example of the structure of an OVF descriptor with the top-level Envelope element follows:



```
<?xml version="1.0" encoding="UTF-8"?>
<Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:vssd="http://schemas.dmtf.org/wbem/wscim/1/cim-
schema/2/CIM_VirtualSystemSettingData"
    xmlns:rasd="http://schemas.dmtf.org/wbem/wscim/1/cim-
schema/2/CIM_ResourceAllocationSettingData"
    xmlns:ovf="http://schemas.dmtf.org/ovf/envelope/1"
    xmlns="http://schemas.dmtf.org/ovf/envelope/1"
    xml:lang="en-US">
    <References>
      <File ovf:id="de-DE-resources.xml" ovf:size="15240"
            ovf:href="http://mywebsite/virtualappliances/de-DE-resources.xml"/>
      <File ovf:id="file1" ovf:href="vmdisk1.vmdk" ovf:size="180114671"/>
      <File ovf:id="file2" ovf:href="vmdisk2.vmdk" ovf:size="4882023564"
ovf:chunkSize="2147483648"/>
      <File ovf:id="file3" ovf:href="resource.iso" ovf:size="212148764"
ovf:compression="gzip"/>
      <File ovf:id="icon" ovf:href="icon.png" ovf:size="1360"/>
    </References>
    <!-- Describes meta-information about all virtual disks in the package -->
    <DiskSection>
        <Info>Describes the set of virtual disks</Info>
        <!-- Additional section content -->
    </DiskSection>
    <!-- Describes all networks used in the package -->
    <NetworkSection>
         <Info>List of logical networks used in the package</Info>
        <!-- Additional section content -->
    </NetworkSection>
    <SomeSection ovf:required="false">
        <Info>A plain-text description of the content</Info>
        <!-- Additional section content -->
```

● There are three stages to exporting and importing virtual machines and templates:

1)Export the virtual machine or template to an export domain.

2)Detach the export domain from one data center, and attach it to another. You can attach it to a different data center in the same Red Hat Enterprise Virtualization environment, or attach it to a data center in a separate Red Hat Enterprise Virtualization environment that is managed by another installation of the Red Hat Enterprise Virtualization Manager.

3)Import the virtual machine or template into the data center to which the export domain is attached.

There are several problems related to Export domain storage.

● **scalability**

Export domain storage is suitable for small data center. But for large datacenter large storage is needed to store the ovfs and image clone in export storage.

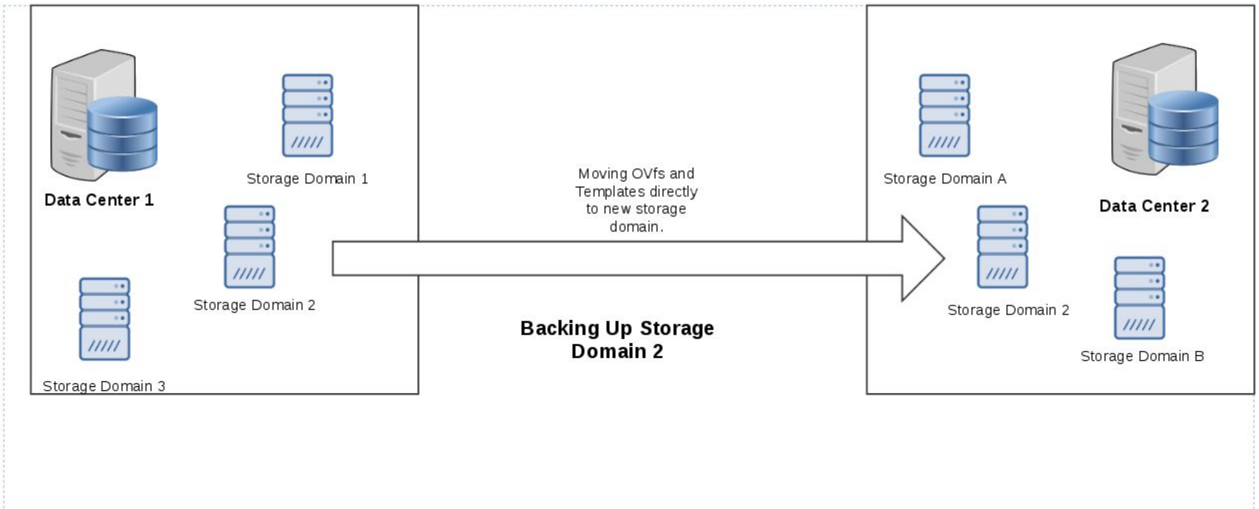
●Large migration time

The migration of entities is done by two copy operations, one to the mediator (Export Storage Domain) and another one from the Export Storage Domain to the target Storage Domain.

●Single point of failure

If export domain storage fail or damage then migration of data to another storage domain is not possible.

Solution



The possible solution for these problems is instead of a dedicated storage domain, use any storage domain as a backup storage.This backup storage can be used to replace deprecated Export storage domain.

Beside that

● It Will help oVirt to achieve an easier alternative solution for the

users to backup entities.

● It Help different platform to use their storage domain as backup.

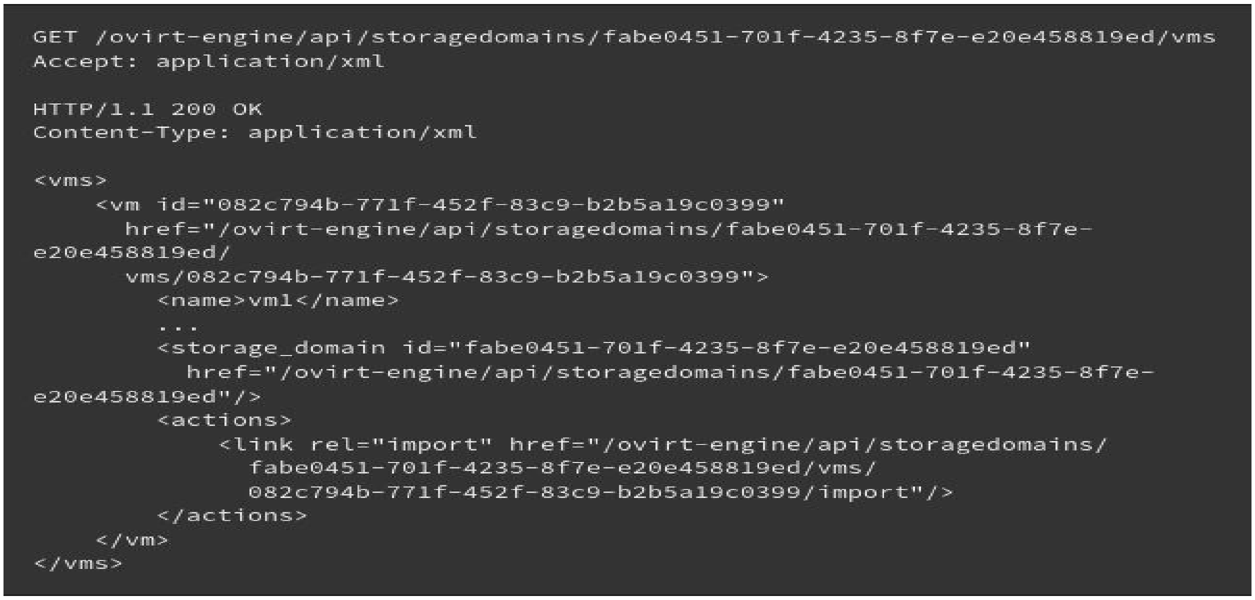
● Make backup migration cost effective only one time coping/cloning.

● You can use multiple storage domain as a backup storage.

Workflow

The new implemented data Storage domain will be the part of ovirt-engine and can be of any defined storage type.Below given steps needed to be implemented in ovirt-engine code base.

Step 1: Selecting the Vms and templates from given storage for backup.



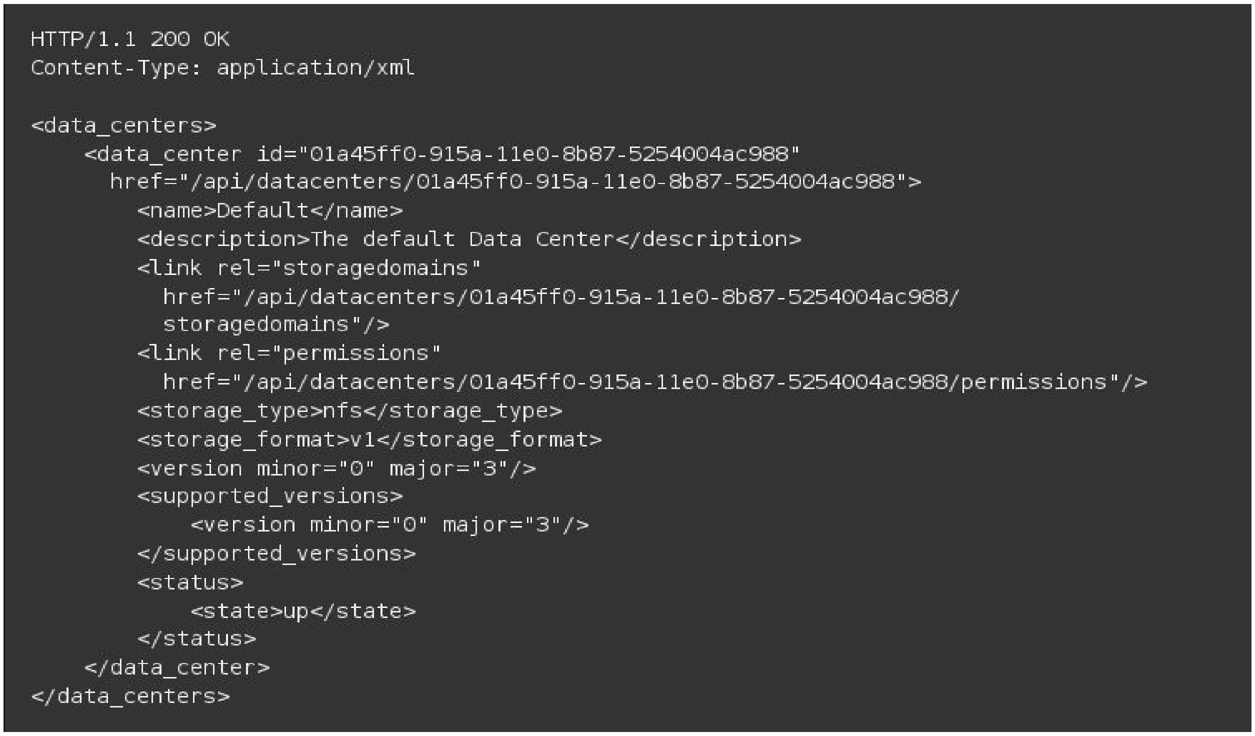
```
GET /ovirt-engine/api/storagedomains/fabe0451-701f-4235-8f7e-e20e458819ed/vms
Accept: application/xml

HTTP/1.1 200 OK
Content-Type: application/xml

<vms>
    <vm id="082c794b-771f-452f-83c9-b2b5a19c0399"
      href="/ovirt-engine/api/storagedomains/fabe0451-701f-4235-8f7e-
e20e458819ed/
      vms/082c794b-771f-452f-83c9-b2b5a19c0399">
        <name>vm1</name>
        ...
        <storage_domain id="fabe0451-701f-4235-8f7e-e20e458819ed"
          href="/ovirt-engine/api/storagedomains/fabe0451-701f-4235-8f7e-
e20e458819ed"/>
        <actions>
            <link rel="import" href="/ovirt-engine/api/storagedomains/
              fabe0451-701f-4235-8f7e-e20e458819ed/vms/
              082c794b-771f-452f-83c9-b2b5a19c0399/import"/>
        </actions>
    </vm>
</vms>
```

Fig 2-Given example showing all available vms in a storage domain.

Step 2: Creating a new Data Storage domain for backup



```
HTTP/1.1 200 OK
Content-Type: application/xml

<data_centers>
    <data_center id="01a45ff0-915a-11e0-8b87-5254004ac988"
      href="/api/datacenters/01a45ff0-915a-11e0-8b87-5254004ac988">
        <name>Default</name>
        <description>The default Data Center</description>
        <link rel="storagedomains"
          href="/api/datacenters/01a45ff0-915a-11e0-8b87-5254004ac988/
          storagedomains"/>
        <link rel="permissions"
          href="/api/datacenters/01a45ff0-915a-11e0-8b87-5254004ac988/permissions"/>
        <storage_type>nfs</storage_type>
        <storage_format>v1</storage_format>
        <version minor="0" major="3"/>
        <supported_versions>
            <version minor="0" major="3"/>
        </supported_versions>
        <status>
            <state>up</state>
        </status>
    </data_center>
</data_centers>
```

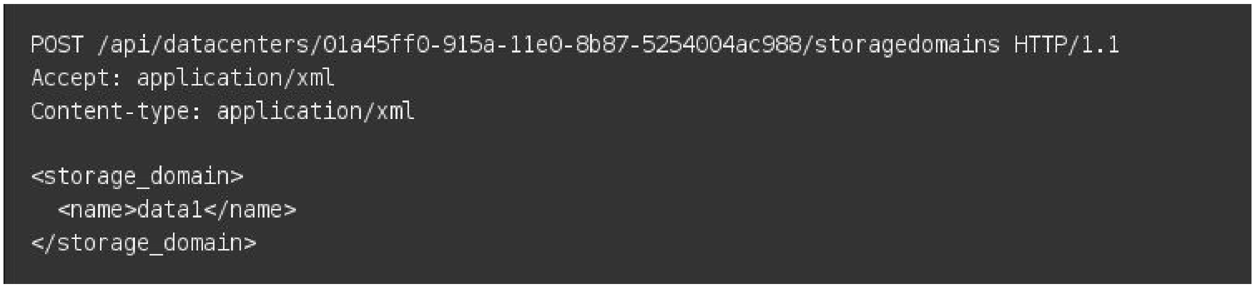
Fig 3- Given example show Rest API request to create a new data storage domain.

Step 3- Once the storage domain is configured as backup the engine will block any running VMs or any changes that might be in the storage domain.

Step 4: Clone the OVfs and templates from source storage domain to destination storage.

- We will clone *OVF_STORE* disk to the storage domain.
- When cloning a Storage Domain, all the entities (VMs, Templates) from the OVF_STORE disk should be retrieved from the tar file and into the DataBase table *unregistered_ovf_of_entities*, later the user can decide how to register them into the new Storage domain's Data Center.

Step 5: Attach this new storage domain to any Data center.

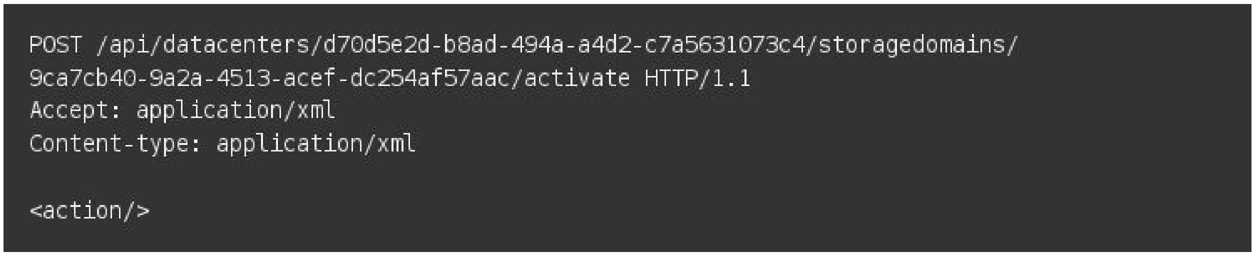


```
POST /api/datacenters/01a45ff0-915a-11e0-8b87-5254004ac988/storagedomains HTTP/1.1
Accept: application/xml
Content-type: application/xml

<storage_domain>
  <name>data1</name>
</storage_domain>
```

Fig 4- Example request to attach a storage domain to data center.

Step 6: Activate the new storage domain.



```
POST /api/datacenters/d70d5e2d-b8ad-494a-a4d2-c7a5631073c4/storagedomains/
9ca7cb40-9a2a-4513-acef-dc254af57aac/activate HTTP/1.1
Accept: application/xml
Content-type: application/xml

<action/>
```

Fig 5-Example request to activate a storage domain.

Adding unregistered floating disk to storage domain:-

- Since floating disks are not part of any VM/Template, the user can register floating disks explicitly.
    - Search for unregistered floating disks in a Storage Domain
    - Get a list of all the unregistered VMs or unregistered Templates by adding the prefix *;unregistered* after the VMs/Templates in the Storage Domain.
- Register an unregistered disk

```
POST
/api/storagedomains/60cec75d-f01d-44a0-9c75-8b415547bc3d/disks;unregistered
HTTP/1.1
Accept: application/xml
Content-type: application/xml
<disk id='8ddb988f-6ab8-4c19-9ea0-b03ab3035347'></disk>
```

Request to register the floating disk to cluster

Implementation

Database

To implement the feature to select any storage as the backup storage we need set the type in storage domain database detail to *backup_storage*.

```
CREATE OR REPLACE FUNCTION Insertstorage_domain_static (
      v_id UUID,
      v_storage VARCHAR(250),
      v_storage_name VARCHAR(250),
      v_storage_description VARCHAR(4000),
      v_storage_comment TEXT,
      v_storage_type INT,
      v_storage_domain_type INT,
      v_storage_domain_format_type VARCHAR(50),
      v_last_time_used_as_master BIGINT,
      v_wipe_after_delete BOOLEAN,
      v_discard_after_delete BOOLEAN,
      v_first_metadata_device VARCHAR(100),
      v_vg_metadata_device VARCHAR(100),
      v_warning_low_space_indicator INT,
      v_critical_space_action_blocker INT
      )
```

For the given database entries *v_storage_domain_type* will be set to the int value which denotes *backup_storage* type.

Coding flow

Note: I have used code snippet from ovirt codebase for better understanding of process and to make correlation from the already available feature. .

The implementation will take place in several steps.The user need to select the storage domain for the backup.

```
public GetExistingStorageDomainListQuery(P parameters) {
      super(parameters);
      }
```

Given function will return all the existing storage domain whereas the storage domain details will be save in `HashSet<Guid> guidsFromDb` .

```
for (StorageDomain domain : domainsInDb) {
                  guidsFromDb.add(domain.getId());
                  }
```

From the retrieved storage domain user will select which storage domain need to be backup.

Before moving further it is needed to check whether the storage domain is empty or have some vms, templates on it because there will be nothing to backup if the storage domain is empty.

```
public GetAllVmsQueryBase(P parameters) {
        super(parameters);
      }
```

And

```
   public GetAllVmTemplatesQuery(P parameters) {
      super(parameters, VmEntityType.TEMPLATE);
      }
```

Will return all the vms present in the storage domain ,if the returned vms list

` List<VM> vmsList = getVMs();` will be empty then we did not need to proceed further.

Once we get the vms and disks list then we can select from them to backup. The selected vms and templates will be added to the list

`List<VM> SelectedvmsList = selectedVMs();` .

After that the user need to create a new storage domain.The given domain can set to any `v_storage_type` type.

```
 protected void addStorageDomainInDb() {
      TransactionSupport.executeInNewTransaction(() -> {
            StorageDomainStatic storageStaticData =
getStorageDomain().getStorageStaticData();
            storageDomainStaticDao.save(storageStaticData);

            getCompensationContext().snapshotNewEntity(storageStaticData);
            StorageDomainDynamic newStorageDynamic =
                  new StorageDomainDynamic(null,
getStorageDomain().getId(), null);

getReturnValue().setActionReturnValue(getStorageDomain().getId());
            storageDomainDynamicDao.save(newStorageDynamic);
            getCompensationContext().snapshotNewEntity(newStorageDynamic);
            getCompensationContext().stateChanged();
            return null;
      });
      if (getStorageDomain().getStorageDomainType().isDataDomain()) {
            createDefaultDiskProfile();
      }
      }
```

Then we need to set the type for that storage domain to be Backup.

```
if (getStorageDomain().getStorageDomainType() == StorageDomainType.BACKUP
                  && getStorageDomain().getStorageType().isBlockDomain()) {
            addValidationMessageVariable("domainType",
StorageConstants.BACKUP);
            addValidationMessageVariable("storageTypes",
StorageConstants.FILE);
            return
      }
```

Once we have set the backup storage and main storage domain then we need to get the ovfs from storage domain to backup storage domain.

```
public GetVmsFromStorageDomainQuery(P parameters, EngineContext
engineContext) {
      super(parameters, engineContext);
      }
```

```
 protected List<VM> buildFromOVFs(List<String> ovfList) {
      List<VM> vms = new ArrayList<>();

      for (String ovf : ovfList) {
            try {
                  if (!ovfHelper.isOvfTemplate(ovf)) {
                  vms.add(ovfHelper.readVmFromOvf(ovf));
                  }
            } catch (OvfReaderException ex) {
                  auditLogOvfLoadError(ex.getName(), ex.getMessage());
            }
      }
```

Vm templates can also be copied in same way.

```
 protected Map<VmTemplate, List<DiskImage>> buildFromOVFs(List<String>
ovfList) {
      Map<VmTemplate, List<DiskImage>> templateDisksMap = new HashMap<>();
      for (String ovf : ovfList) {
            try {
                  if (ovfHelper.isOvfTemplate(ovf)) {
                  VmTemplate vmTemplate =
ovfHelper.readVmTemplateFromOvf(ovf);
                  List<DiskImage> templateDisks = new
ArrayList<>(vmTemplate.getDiskTemplateMap().values());
                  templateDisksMap.put(vmTemplate, templateDisks);
                  }
            } catch (OvfReaderException ex) {
                  auditLogOvfLoadError(ex.getName(), ex.getMessage());
            }
      }
```

Once the ovfs and templates are copied to backup storage domain then we can attach the storage domain to any existing data center.

```
 protected void addStorageDomain() {
      initializeStorageDomain();
      addStorageDomainInDb();
      // check connection to storage
      Pair<Boolean, Integer> connectReturnValue = connectStorage();
      if (!connectReturnValue.getFirst()) {
            EngineFault fault = new EngineFault();
```

```
fault.setError(EngineError.forValue(connectReturnValue.getSecond()));
            getReturnValue().setFault(fault);
            setSucceeded(false);
      }
      else if (addStorageDomainInIrs()) {
            updateStorageDomainDynamicFromIrs();
            setSucceeded(true);
      }
      }
```

Later we can activate the new storage domain.The storage domain can be used as data domain after the activation.

Timeline

1) **Exams Period [**27 Apr to 4 May**]**

I got my exams in this period so will not be active in that period but it is before the start of project.

2) **Community Bonding Period [**5 May to 30 May**]**

❏ My summer vacations are starting from 5th May.
❏ Since I have already read the documentation I will not spent the time on that.
❏ Instead try to get familiarize with the code base of Ovirt and Ovirt-engine.
❏ Get to be in contact with mentor to discuss the implementation.

3) **Week 1-2 [**31 May to 10 June**]**

❏ Look for the changes need to be done in the destination backup storage domain.
❏ Start to implement those changes.

4) **Week 3 [**11 June to 18 June**]**

❏ Complete the changes in destination backup storage.
❏ Look for the changes required in database of ovirt.

5) **Week 4-Till Phase 1 Evaluation [**19 June to 26 June**]**

❏ Try to get familiar with gerrit code review.
❏ Start the testing of code with lago.
❏ Submit the patch created till that time.

6)**Week 6 [**27 June to 3 July**]**

❏ Code the changes need to be done in ovirt-engine database

❏ Implement code to block all the vms while having backup.

7)**Week 7 [**3 July to 11 July**]**

❏ Look if any new cloning method is required or already built in method need to be use.
❏ Implement the cloning/Backup method for new storage domain.

8)**Week 8-Till phase 2[**12 July to 27 July**]**

❏ Testing of code till now.
❏ Create the new api request if required.

9)**Week 10-11[**28 July to 5 August**]**

❏ Tried to complete any other change required in ovirt-engine.

10)**Week 12-Final Week [**6 August to 21 August **]**

❏ Complete all other changes required.
❏ Do the documentation of new storage domain feature.
❏ Do the final testing of project changes using Lago.

I don't have any pre plan in this summer so after starting of my holiday I will give about 40-45 hours a week. If require then I can spent more time on project to maintain the pace of work. My college will open again on 28th June. After that I will decrease the amount of time spent to 30 hours a week.I got no other exam or other activity in the middle of whole gsoc period but still I will try to work to complete the project 2 or 3 week earlier so that if I get on some holiday for a while then I can catch up that later.Even if I manage to complete the project earlier then I would look for the option work to be done.

About me

I am Shubham Dubey. I am a 3rd year Undergraduate Student of Computer Science at The LNM Institute of Information Technology,India.My area of work and interest is in field of Cloud Computing ,Virtualization and Security(Reverse engineering and Malware Analysis).I am using linux and other open source software since last 5 or 6 years.I have authored few Articles on linux related topics in OpenSource4you magazine[OpenSourceForU articles].I have done RHCE v6 and v7 and RHOA(Red Hat OpenStack Administration) course just after passing high school.I spent most of my time in learning new technologies in Linux or in Security related work.

Motivation for Gsoc

The biggest motivation for apply for gsoc and starting contribution on open source project is to make those software better for use for myself and other. I find lots of good open source alternative or core software and feel that they need to be maintained continuously. For that every user need to contribute on some project to maintain the standard of open source tools.To start the contribution I think gsoc is the best place to get started.

Applying on ovirt project is been my prefered choice since I have experience and interest in virtualization softwares. Another profit of applying for ovirt is my past experience in openstack contribution. It helped me to understand the technology,method used in ovirt.Till now I get good response from the Community, they helped me a lot in my query and I would love to continue my contribution on ovirt after the gsoc period also.

Relevant Open Source Projects and Contribution

1. Worked on Cinder incremental backup bug

- Cinder was forcing full backup instead of incremental backup on ceph storage.
- Problem causing due to Ceph differential backup test fail.

2.Developed own Cloud as Intern at Linux World Pvt Ltd

- Created cloud having Iaas, Staas and Saas services.
- Based on python as backend and web based frontend.
- Use technology like docker and iSCSI.
- Used docker container for giving software as service to user.
- Used iSCSI for block storage and sparse files, NFS for file storage.

3.BigD Music player[link]

- Java based music player with visualization support.
- Used swing as GUI.

● Support feature like creating playlist, internet radio and many more.

4.Smart Class Web Application[link]

● A Python Django Framework based web application for student and teachers online interaction .

● Helps in resource sharing, General notification and online checking system.

5.XML Remote Connector[link]

● Remote server client application that can use for doing any simple task

in remote server which is listening for some connection.

● Use XML rpc based API.

● Can do the work of ssh,scp and nc at same time.

I have created some more python based utility and scripts which you can find at GitHub.[GitHub profile]

References and Further Reads

❖ Exporting and importing Virtual Machines and Templates [link]
❖ Import Storage Domain [link]
❖ Rest API Guide [link]
❖ Storage in Ovirt [link]
❖ Disaster Recovery in Ovirt [link]