

在Cinder创建Volume时，cinder\_api会创建一个名为“create\_volume\_api”的Flow，该Flow中的ExtractVolumeRequestTask，会对传入的Zone名称进行过滤，如果过滤结果与预期不符，创建Volume将会失败。

cinder.volume.flows.api.create\_volume.ExtractVolumeRequestTask的execute方法：

```
def execute(self, context, size, snapshot, image_id, source_volume,
            availability_zone, volume_type, metadata, key_manager,
            source_replica, consistencygroup, cgsnapshot, group):

    .....

    availability_zone = self._extract_availability_zone(availability_zone,
                                                         snapshot,
                                                         source_volume,
                                                         group)

    .....
```

从当前的日志分析：ExtractVolumeRequestTask中的\_extract\_availability\_zone抛出异常，导致ExtractVolumeRequestTask执行失败，整个创建Vol流

```
node2 2017-05-09 10:24:49.590 24 INFO cinder.api.openstack.wsgi [req-f4992119-6da5-44e1-b270-a41ff1def209 - - - -] POST http://172.24.9.198:8776/v2/406cd353135e44f0ade98f53d92d5d8b/volumes
node2 2017-05-09 10:24:49.591 24 INFO cinder.api.v2.volumes [req-f4992119-6da5-44e1-b270-a41ff1def209 - - - -] Create volume of 10 GB
node2 2017-05-09 10:24:49.649 24 INFO cinder.volume.api [req-f4992119-6da5-44e1-b270-a41ff1def209 - - - -] Availability Zones retrieved successfully.
node2 2017-05-09 10:24:49.728 24 WARNING cinder.volume.api [req-f4992119-6da5-44e1-b270-a41ff1def209 - - - -] Task 'cinder.volume.flows.api.create_volume.ExtractVolumeRequestTask;volume:create'
(d44d022f-d2f0-4ee8-865c-de39dc9a483a) transitioned into state 'FAILURE' from state 'RUNNING'
node2 Failure: cinder.exception.InvalidInput: Invalid input received: Availability zone 'nova_ctl' is invalid.
node2 2017-05-09 10:24:49.734 24 WARNING cinder.volume.api [req-f4992119-6da5-44e1-b270-a41ff1def209 - - - -] Task 'cinder.volume.flows.api.create_volume.ExtractVolumeRequestTask;volume:create'
(d44d022f-d2f0-4ee8-865c-de39dc9a483a) transitioned into state 'REVERTED' from state 'REVERTING'
node2 2017-05-09 10:24:49.736 24 WARNING cinder.volume.api [req-f4992119-6da5-44e1-b270-a41ff1def209 - - - -] Flow 'volume_create_api' (82075ad4-77fc-4ab0-a3ab-6116c7e72845) transitioned into state
'REVERTED' from state 'RUNNING'
node2 2017-05-09 10:24:49.737 24 INFO cinder.api.openstack.wsgi [req-f4992119-6da5-44e1-b270-a41ff1def209 - - - -] http://172.24.9.198:8776/v2/406cd353135e44f0ade98f53d92d5d8b/volumes returned with HTTP 400
node2 2017-05-09 10:24:49.738 24 INFO eventlet.wsgi.server [req-f4992119-6da5-44e1-b270-a41ff1def209 - - - -] 172.24.9.12,172.24.9.13 "POST /v2/406cd353135e44f0ade98f53d92d5d8b/volumes HTTP/1.1" status:
400 len: 371 time: 0.1560950
```

分析\_extract\_availability\_zone方法的代码：

```
def _extract_availability_zone(self, availability_zone, snapshot,
                              source_volume, group):

    """Extracts and returns a validated availability zone.

    This function will extract the availability zone (if not provided) from
    the snapshot or source_volume and then performs a set of validation
    checks on the provided or extracted availability zone and then returns
    the validated availability zone.
    """

    # If the volume will be created in a group, it should be placed in
    # in same availability zone as the group.
    """

    group指的是cinder的consistency groups特性（https://docs.openstack.org/admin-guide/blockstorage-consistency-groups.html）
    该参数传入值当前情况下一般为None
    """

    if group:
        try:
            availability_zone = group['availability_zone']
        except (TypeError, KeyError):
            pass

    # Try to extract the availability zone from the corresponding snapshot
    # or source volume if either is valid so that we can be in the same
    # availability zone as the source.
    """

    如果不指定创建卷时的AZ，那么：
    1. 从快照创建时，指定为快照的AZ，从已有卷创建时，指定为已有卷AZ；
    2. 如果方案1任然不能指定出AZ，那么依次取default_availability_zone，storage_availability_zone
       默认情况下default_availability_zone的值为None；
       而storage_availability_zone的值为Nova
    """

    if availability_zone is None:
        if snapshot:
```

```

    try:
        availability_zone = snapshot['volume']['availability_zone']
    except (TypeError, KeyError):
        pass
    if source_volume and availability_zone is None:
        try:
            availability_zone = source_volume['availability_zone']
        except (TypeError, KeyError):
            pass

    if availability_zone is None:
        if CONF.default_availability_zone:
            availability_zone = CONF.default_availability_zone
        else:
            # For backwards compatibility use the storage_availability_zone
            availability_zone = CONF.storage_availability_zone

```

从环境上打印的信息来看self.availability\_zones里只包含“nova”一个AZ，allow\_availability\_zone\_fallback而值为False导致了抛出异常。

```

"""
从环境上打印的信息来看self.availability_zones里只包含“nova”一个AZ，
allow_availability_zone_fallback而值为False导致了抛出异常。
"""

if availability_zone not in self.availability_zones:
    """
    allow_availability_zone_fallback的作用是，当输入的AZ不满足情形时，对输入进行回退！默认为False！！
    先回退为default_availability_zone或者storage_availability_zone（default_availability_zone优先级更高）！！
    默认情况下是回退为“nova”的！！
    """

    if CONF.allow_availability_zone_fallback:
        original_az = availability_zone
        availability_zone = (
            CONF.default_availability_zone or
            CONF.storage_availability_zone)
        LOG.warning(_LW("Availability zone '%(s_az)s' "
                        "not found, falling back to "
                        "'%(s_fallback_az)s'. "),
                    {'s_az': original_az,
                     's_fallback_az': availability_zone})
    else:
        msg = _("Availability zone '%(s_az)s' is invalid.")
        msg = msg % {'s_az': availability_zone}
        raise exception.InvalidInput(reason=msg)

# If the configuration only allows cloning to the same availability
# zone then we need to enforce that.
"""
cloned_volume_same_az表示创建卷时是否要求源的AZ和输入AZ一致（主要针对从快照创建和卷创建），默认为True!!
"""

if CONF.cloned_volume_same_az:
    snap_az = None
    try:
        snap_az = snapshot['volume']['availability_zone']
    except (TypeError, KeyError):
        pass
    if snap_az and snap_az != availability_zone:
        msg = _("Volume must be in the same "
                "availability zone as the snapshot")
        raise exception.InvalidInput(reason=msg)
    source_vol_az = None
    try:
        source_vol_az = source_volume['availability_zone']
    except (TypeError, KeyError):
        pass

```

```

        if source_vol_az and source_vol_az != availability_zone:
            msg = _("Volume must be in the same "
                    "availability zone as the source volume")
            raise exception.InvalidInput(reason=msg)

    return availability_zone

```

从上面的分析可以看出问题的关键在于self.availability\_zones中不包含输入的AZ

回到create\_volume.py中可以看到ExtractVolumeRequestTask在get\_flow方法中创建:

```

def get_flow(db_api, image_service_api, availability_zones, create_what,
             scheduler_rpcapi=None, volume_rpcapi=None):
    .....
    api_flow.add(ExtractVolumeRequestTask(
        image_service_api,
        availability_zones,
        rebind={'size': 'raw_size',
                'availability_zone': 'raw_availability_zone',
                'volume_type': 'raw_volume_type'}))
    .....
    return taskflow.engines.load(api_flow, store=create_what)

```

其入参availability\_zones在cinder.volume.api.API的create方法和list\_availability\_zones方法中组装,

create方法:

```

raw_zones = self.list_availability_zones(enable_cache=True)
availability_zones = set([az['name'] for az in raw_zones])
if CONF.storage_availability_zone:
    availability_zones.add(CONF.storage_availability_zone)

```

list\_availability\_zones方法:

```

def list_availability_zones(self, enable_cache=False):
    """Describe the known availability zones

    :retval tuple of dicts, each with a 'name' and 'available' key
    """
    refresh_cache = False
    if enable_cache:
        #self.availability_zones_last_fetched表示上次更新cache时间
        if self.availability_zones_last_fetched is None:
            refresh_cache = True
        else:
            cache_age = timeutils.delta_seconds(
                self.availability_zones_last_fetched,
                timeutils.utcnow())
            #az_cache_duration默认是一小时
            if cache_age >= CONF.az_cache_duration:
                refresh_cache = True
    if refresh_cache or not enable_cache:
        topic = constants.VOLUME_TOPIC          #cinder-volume, 注意这里一些旧的服务的AZ也会被带进来, 只要没有disabled
        ctxt = context.get_admin_context()
        services = objects.ServiceList.get_all_by_topic(ctxt, topic)
        az_data = [(s.availability_zone, s.disabled)
                    for s in services]
        disabled_map = {}
        for (az_name, disabled) in az_data:
            tracked_disabled = disabled_map.get(az_name, True)
            disabled_map[az_name] = tracked_disabled and disabled
        azs = [{'name': name, 'available': not disabled}
                for (name, disabled) in disabled_map.items()]
    if refresh_cache:

```

```

        now = timeutils.utcnow()
        self.availability_zones = azs
        self.availability_zones_last_fetched = now
        LOG.debug("Availability zone cache updated, next update will"
                  " occur around %s.", now + datetime.timedelta(
                      seconds=CONF.az_cache_duration))
    else:
        azs = self.availability_zones
    LOG.info(_LI("Availability Zones retrieved successfully."))
    return tuple(azs)

```

基本可以确定：ExtractVolumeRequestTask. availability\_zones的值来自于数据库的services表！！

通过在volume节点配置storage\_availability\_zone可以修改！！

```
[stack@localhost logs]$ cinder service-list
```

Binary	Host	Zone	Status	State	Updated_at	Disabled Reason
cinder-scheduler	localhost.localdomain	nova	enabled	up	2017-05-08T07:25:37.000000	-
cinder-volume	localhost.localdomain@NetAppIscsiBackend	nova	enabled	up	2017-05-08T07:25:32.000000	-
cinder-volume	localhost.localdomain@NetAppIscsiBackend_https	nova	enabled	down	2017-05-04T06:52:49.000000	-
cinder-volume	localhost.localdomain@lvmdriver-1	nova	enabled	down	2017-05-04T06:52:38.000000	-