
n0stack Documentation

h-otter

May 04, 2019

User Documentation

1	Description	3
2	Motivation	5

The n0stack is a simple cloud provider using gRPC.

The n0stack is...

- **a cloud provider.**
 - You can use some features: booting VMs, managing networks and so on (see also /n0proto.)
- **simple.**
 - There are shortcode and fewer options.
- **using gRPC.**
 - A unified interface increase reusability.
- **able to be used as library and framework.**
 - You can concentrate to develop your logic by sharing libraries and frameworks for middleware, test, and deployment.

Cloud providers have various forms depending on users. This problem has been solved with many options and add-ons (e.g. OpenStack configuration file is very long.) However, it is difficult to adapt to the application by options, then it is necessary to read or rewrite long abstracted codes. Therefore, I thought that it would be better to code on your hands from beginning.

There are some problems to develop cloud providers from scratch: no library, software quality, man-hour, and deployment. The n0stack wants to solve such problems.

2.1 Quick Start

2.1.1 n0cli

The n0cli is a CLI tool to call n0stack gRPC APIs.

Installation

with docker

```
docker pull n0stack/n0stack
docker run -it --rm -v /usr/local/bin:/dst n0stack/n0stack cp /usr/local/bin/n0cli /
↳dst/
```

Usage

- See also command help.

```
$ n0cli --api-endpoint=$api_ip:20180 get node
{
  "nodes": [
```

(continues on next page)

```
{
  "name": "vm-host1",
  "annotations": {
    "github.com/n0stack/n0stack/n0core/agent_version": "52"
  },
  "address": "192.168.122.10",
  "serial": "Specified",
  "cpu_milli_cores": 1000,
  "memory_bytes": "1033236480",
  "storage_bytes": "107374182400",
  "unit": 1,
  "state": "Ready",
  "reserved_computes": {
    "debug_ipv6": {
      "annotations": {
        "n0core/provisioning/virtual_machine/virtual_machine/reserved_by": "debug_
↪ipv6"
      },
      "request_cpu_milli_core": 10,
      "limit_cpu_milli_core": 1000,
      "request_memory_bytes": "536870912",
      "limit_memory_bytes": "536870912"
    }
  },
  "reserved_storages": {
    "debug-ipv6-network": {
      "annotations": {
        "n0core/provisioning/block_storage/reserved_by": "debug-ipv6-network"
      },
      "request_bytes": "1073741824",
      "limit_bytes": "10737418240"
    },
    "debug_ipv6_network": {
      "annotations": {
        "n0core/provisioning/block_storage/reserved_by": "debug_ipv6_network"
      },
      "request_bytes": "1073741824",
      "limit_bytes": "10737418240"
    },
    "ubuntu-1804": {
      "annotations": {
        "n0core/provisioning/block_storage/reserved_by": "ubuntu-1804"
      },
      "request_bytes": "1073741824",
      "limit_bytes": "10737418240"
    }
  }
}
```

Examples

See also Usecases.

2.2 Overview about n0proto

The n0proto is gRPC definitions for all of n0stack API.

2.2.1 Resources

Budget

Budget define data structure about resource budget: CPU, Memory, IP address, MAC address, storage, and so on.

Budget CPU

Pool

Pool ensure Budgets.

Pool Budget

Node

-
- CPU

Network

-
- IPMAC

Provisioning

Provisioning create virtual resources on ensured budget.

Pool

BlockStorage

- NodeStorage
 - Qcow2
1. NodeReserveStorage / ScheduleStorage
 2. Node

VirtualMachine

- NodeCompute(CPU)VM
 - BlockStorageNetworkNetworkInterface(MACIP)
1. NodeReserveCompute / ScheduleComputeCPU
 2. BlockStorageSetInuseBlockStorage
 3. NetworkReserveNetworkInterfaceMACIP
 4. Node

Deployment

Deployment abstract Provisioning operations.

DeploymentProvisioning

Image

- BlockStorage
- ImageBlockStorageBlockStorage
- ImageBlockStorageBlockStorageVMOpenstackGlance ([detail](#))

2.2.2 Naming Conventions about API

Standard fields

name

- Unique key

annotations

- Field that stores implementation-dependent values

[Resource type]_name

- Reference to resources

*s

- List field

Standard methods

List

Get

Create / Apply

Update

Delete

2.3 Usecases

2.3.1 List

Boot VirtualMachine from Image

In case of booting VirtualMachine test with Image cloudimage-ubuntu tagged 18.04 on Network test-network.

(If you don't have registered Image cloudimage-ubuntu tagged 18.04, refer [here](#) around FetchISO, ApplyImage and RegisterBlockStorage tasks.)

Example

```

GenerateBlockStorage:
  type: Image
  action: GenerateBlockStorage
  args:
    image_name: cloudimage-ubuntu
    tag: "18.04"
    block_storage_name: test-blockstorage
    annotations:
      n0core/provisioning/block_storage/request_node_name: vm-host1
    request_bytes: 1073741824
    limit_bytes: 1073741824

ApplyNetwork:
  type: Network
  action: ApplyNetwork
  args:
    name: test-network
    ipv4_cidr: 192.168.0.0/24
    annotations:
      n0core/provisioning/virtual_machine/vlan_id: "100"

CreateVirtualMachine:
  type: VirtualMachine
  action: CreateVirtualMachine
  args:

```

(continues on next page)

(continued from previous page)

```

name: test-vm
annotations:
  n0core/provisioning/virtual_machine/request_node_name: vm-host1
request_cpu_milli_core: 10
limit_cpu_milli_core: 1000
request_memory_bytes: 536870912
limit_memory_bytes: 536870912
block_storage_names:
  - test-blockstorage
nics:
  - network_name: test-network
    ipv4_address: 192.168.0.1
uuid: 056d2ccd-0c4c-44dc-a2c8-39a9d394b51f
# cloud-config related options:
login_username: n0user
ssh_authorized_keys:
  - ecdsa-sha2-nistp256_
↪AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBITowPn2O1leCvXN5XV+Lb6jfXzgDbXyEdtadayadDUJtFrcl
↪testkey
depends_on:
  - GenerateBlockStorage
  - ApplyNetwork

```

```
n0cli --api-endpoint=$api_ip:20180 do $path_of_previous_yaml
```

Then, you can login virtual machine via ssh by n0user user using key below:

```

-----BEGIN EC PRIVATE KEY-----
MHcCAQEEIBAQh+adEg/rjqj9qLE0jI4EqV8kZFDzWTASAwvx6HWdoAoGCCqGSM49
AwEHoUQDQgAEhOjA+fY6XV4K9c3ldX4tvqN9fOANTfIR21rJp0NQm0Wtw3abaaML
UHbRUECglxm1JiSaOuWVLTpDbpN7mxNi8Q==
-----END EC PRIVATE KEY-----

```

(Ubuntu 18.04 Cloud Image doesn't allow password login to ssh configured above, so you need set password if need to access via VNC console)

Overview

- Image BlockStorage
 - Image Docker Image
 - * n0cli get image cloudimage-ubuntu-1804 tags
 - block_storage_name **BlockStorage**
 - * VirtualMachine VM
 - annotations n0core/provisioning/block_storage/request_node_name **Block-Storage**
 - * n0cli get node
 - BlockStorage 10 GB (10737418240 Bytes)
 - * OS
 - BlockStorage 1 GB (1073741824 Bytes)

- *
 - Network /
 - VirtualMachine
 - request_cpu_milli_core CPU limit_cpu_milli_core
 - * limit_cpu_milli_core CPU limit_cpu_milli_core % 1000 == 0
 - * 1VM
 - request_memory_bytes == limit_memory_bytes
 - * 512 MB (536870912 Bytes) VM
 - * KVMmemory ballooning
 - annotations n0core/provisioning/virtual_machine/request_node_name BlockStorage
 - block_storage_names BlockStorage
 - * Image BlockStorage
 - nics Network
 - * Network 192.168.0.1
 - uuid uuidgen
 - OS cloud-init nicsIPlogin_username ssh_authorized_keys SSH

Inverse action

```

Delete_test-vm:
  type: VirtualMachine
  action: DeleteVirtualMachine
  args:
    name: test-vm

Delete_test-blockstorage:
  type: BlockStorage
  action: DeleteBlockStorage
  args:
    name: test-blockstorage
  depends_on:
    - Delete_test-vm

Delete_test-network:
  type: Network
  action: DeleteNetwork
  args:
    name: test-network
  depends_on:
    - Delete_test-vm

```

Tips: Idempotent action

Caution: This DAG deletes block storage and VM which you created, often causes misoperation **unintentionally**.

```
Delete_test-vm:
  type: VirtualMachine
  action: DeleteVirtualMachine
  args:
    name: test-vm
  ignore_error: true

Delete_test-blockstorage:
  type: BlockStorage
  action: DeleteBlockStorage
  args:
    name: test-blockstorage
  depends_on:
    - Delete_test-vm
  ignore_error: true

Delete_test-network:
  type: Network
  action: DeleteNetwork
  args:
    name: test-network
  depends_on:
    - Delete_test-vm
  ignore_error: true

GenerateBlockStorage:
  type: Image
  action: GenerateBlockStorage
  args:
    image_name: cloudimage-ubuntu
    tag: "18.04"
    block_storage_name: test-blockstorage
    annotations:
      n0core/provisioning/block_storage/request_node_name: vm-host1
    request_bytes: 1073741824
    limit_bytes: 1073741824
  depends_on:
    - Delete_test-blockstorage

ApplyNetwork:
  type: Network
  action: ApplyNetwork
  args:
    name: test-network
    ipv4_cidr: 192.168.0.0/24
    annotations:
      n0core/provisioning/virtual_machine/vlan_id: "100"
  depends_on:
    - Delete_test-network

CreateVirtualMachine:
  type: VirtualMachine
  action: CreateVirtualMachine
  args:
    name: test-vm
    annotations:
      n0core/provisioning/virtual_machine/request_node_name: vm-host1
```

(continues on next page)

(continued from previous page)

```

request_cpu_milli_core: 10
limit_cpu_milli_core: 1000
request_memory_bytes: 536870912
limit_memory_bytes: 536870912
block_storage_names:
  - test-blockstorage
nics:
  - network_name: test-network
    ipv4_address: 192.168.0.1
uuid: 056d2ccd-0c4c-44dc-a2c8-39a9d394b51f
login_username: n0user
ssh_authorized_keys:
  - ecdsa-sha2-nistp256
↪AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBITowPn2O1leCvXN5XV+Lb6jfXzgDbXyEdt ayadDUJtFrct
↪testkey
depends_on:
  - GenerateBlockStorage
  - ApplyNetwork

```

Boot VirtualMachine from ISO

Fetch and register Ubuntu 18.04 Cloud Images

```

FetchISO:
  type: BlockStorage
  action: FetchBlockStorage
  args:
    name: cloudimage-ubuntu-1804
    annotations:
      n0core/provisioning/block_storage/request_node_name: vm-host1
    request_bytes: 1073741824 # 1GiB
    limit_bytes: 10737418240 # 10GiB
    source_url: https://cloud-images.ubuntu.com/bionic/current/bionic-server-cloudimg-
↪amd64.img

ApplyNetwork:
  type: Network
  action: ApplyNetwork
  args:
    name: test-network
    ipv4_cidr: 192.168.0.0/24
    annotations:
      n0core/provisioning/virtual_machine/vlan_id: "100"

CreateVirtualMachine:
  type: VirtualMachine
  action: CreateVirtualMachine
  args:
    name: test-vm
    annotations:
      n0core/provisioning/virtual_machine/request_node_name: vm-host1
    request_cpu_milli_core: 10
    limit_cpu_milli_core: 1000
    request_memory_bytes: 1073741824 # 1GiB

```

(continues on next page)

(continued from previous page)

```

limit_memory_bytes: 1073741824 # 1GiB
block_storage_names:
  - cloudimage-ubuntu-1804
nics:
  - network_name: test-network
    ipv4_address: 192.168.0.1
  # cloud-config related options:
login_username: n0user
ssh_authorized_keys:
  - ecdsa-sha2-nistp256
→AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBITowPn2O11eCvXN5XV+Lb6jfXzgDbXyEdt ayadDUJtFrcl
→testkey
depends_on:
  - CreateBlockStorage

# You need to set password for user to login via console (not set if default)
OpenConsole:
  type: VirtualMachine
  action: OpenConsole
  args:
    name: test-vm
  depends_on:
    - CreateVirtualMachine

```

Then, you can login virtual machine via ssh by n0user user using key below:

```

-----BEGIN EC PRIVATE KEY-----
MHcCAQEEIBAqh+adEg/rjqj9qLE0jiI4EqV8kZFDzWTASAwvx6HWdoAoGCCqGSM49
AwEHoUQDQgAEhOjA+fY6XV4K9c3ldX4tvqN9fOANt fIR21rJpONQm0Wtw3abaaML
UHbrUECglxml1JiSaOuWVLTpDbpN7mxNi8Q==
-----END EC PRIVATE KEY-----

```

(Ubuntu 18.04 Cloud Image doesn't allow password login to ssh configured above, so you need set password if need to access via VNC console)

Inverse action

```

Delete_test-vm:
  type: VirtualMachine
  action: DeleteVirtualMachine
  args:
    name: test-vm

Delete_blockstorage:
  type: BlockStorage
  action: DeleteBlockStorage
  args:
    name: cloudimage-ubuntu-1804
  depends_on:
    - Delete_test-vm

Delete_test-network:
  type: Network
  action: DeleteNetwork
  args:

```

(continues on next page)

(continued from previous page)

```

name: test-network
depends_on:
  - Delete_test-vm

```

Register blockstorage as an Image

You can manage blockstorages by registering to image, versioning blockstorage with tag.

```

FetchISO:
  type: BlockStorage
  action: FetchBlockStorage
  args:
    name: cloudimage-ubuntu-1804
    annotations:
      n0core/provisioning/block_storage/request_node_name: vm-host1
    request_bytes: 1073741824 # 1GiB
    limit_bytes: 10737418240 # 10GiB
    source_url: https://cloud-images.ubuntu.com/bionic/current/bionic-server-cloudimg-
↳amd64.img

ApplyImage:
  type: Image
  action: ApplyImage
  args:
    name: cloudimage-ubuntu

RegisterBlockStorage:
  type: Image
  action: RegisterBlockStorage
  args:
    image_name: cloudimage-ubuntu
    block_storage_name: cloudimage-ubuntu-1804
    tags:
      - latest
      - "18.04"
  depends_on:
    - ApplyImage

```

Generate BlockStorage from Image

```

GenerateBlockStorage:
  type: Image
  action: GenerateBlockStorage
  args:
    image_name: cloudimage-ubuntu
    tag: "18.04"
    block_storage_name: test-blockstorage
    annotations:
      n0core/provisioning/block_storage/request_node_name: vm-host1
    request_bytes: 1073741824
    limit_bytes: 10737418240

```

Delete image

```
Remove_cloudimage-ubuntu:  
  type: Image  
  action: DeleteImage  
  args:  
    name: cloudimage-ubuntu  
  depends_on:  
    - Delete_test-vm
```

Delete image (detailed)

```
Untag_1804_from_cloudimage-ubuntu:  
  type: Image  
  action: UntagImage  
  args:  
    name: cloudimage-ubuntu  
    tag: "18.04"  
  depends_on:  
    - Delete_test-vm  
  
Untag_latest_from_cloudimage-ubuntu:  
  type: Image  
  action: UntagImage  
  args:  
    name: cloudimage-ubuntu  
    tag: latest  
  depends_on:  
    - Delete_test-vm  
  
Unregister_cloudimage-ubuntu-1804-from-cloudimage-ubuntu:  
  type: Image  
  action: UnregisterBlockStorage  
  args:  
    image_name: cloudimage-ubuntu:  
    block_storage_name: cloudimage-ubuntu-1804  
  depends_on:  
    - Untag_1804_from_cloudimage-ubuntu  
    - Untag_latest_from_cloudimage-ubuntu  
  
Remove_cloudimage-ubuntu:  
  type: Image  
  action: DeleteImage  
  args:  
    name: cloudimage-ubuntu  
  depends_on:  
    - Unregister_cloudimage-ubuntu-1804-from-cloudimage-ubuntu  
  
Remove_cloudimage-ubuntu-1804:  
  type: BlockStorage  
  action: DeleteBlockStorage  
  args:  
    name: cloudimage-ubuntu-1804  
  depends_on:  
    - Unregister_cloudimage-ubuntu-1804-from-cloudimage-ubuntu
```

2.4 Architectural Decision Records

2.4.1 Status

- proposed
- accepted
- deprecated

2.4.2 Translate to English

I will do when I remember. :pray:

2.4.3 List

n0core packages

Status	accepted

Context

-

Decision

n0core/pkg/api

- API

n0core/pkg/datastore

-

n0core/pkg/driver

-

n0core/pkg/util

-

n0core/pkg/deploy

-

n0proto.go/*

- `n0protogRPCmake build-n0proto`

n0proto.go/pkg/transaction

-
- TODO: n0core/pkg/util

Consequences

-

Lock about update process

Status	accepted

Context

- - DB
 - IP
- - TODO:

Decision

- `github.com/n0stack/n0stack/n0core/pkg/datastore/lock`
- `api Datastore.Lock(key string)`
 -
 - datastore
 - * · panic
 - ReserveStorage
 - * Create
 - * Delete
 - *
- -

-
- DBn0core
 - n0core

Consequences

-

Reference

- #115

About pending state

Status	accepted

Context

APIVirtualMachineBlockStorageAPI

Decision

- VirtualMachineBlockStorageCreate PENDING
 - PENDING

API PENDING

Example in BlockStorage

- - PENDING
-

```
func (a *BlockStorageAPI) CheckAndLock(tx *transaction.Transaction, bs *provisioning.
↳BlockStorage) error {
    prev := &provisioning.BlockStorage{}
    if err := a.dataStore.Get(bs.Name, prev); err != nil {
        log.Printf("[WARNING] Failed to get data from db: err='%s'", err.Error())
        return grpcutil.WrapGrpcErrorf(codes.Internal, "Failed to get '%s' from db,
↳please retry or contact for the administrator of this cluster", bs.Name)
    } else if prev.Name != "" {
        return grpcutil.WrapGrpcErrorf(codes.AlreadyExists, "BlockStorage '%s' is
↳already exists", bs.Name)
    }

    bs.State = provisioning.BlockStorage_PENDING
}
```

(continues on next page)

(continued from previous page)

```

    if err := a.dataStore.Apply(bs.Name, bs); err != nil {
        return grpcutil.WrapGrpcErrorf(codes.Internal, "Failed to apply data for db:
↳err='%s'", err.Error())
    }
    tx.PushRollback("free optimistic lock", func() error {
        return a.dataStore.Delete(bs.Name)
    })

    return nil
}

```

- - PENDING
- PENDING
-

```

func (a *BlockStorageAPI) GetAndLock(tx *transaction.Transaction, name string)
↳(*pprovisioning.BlockStorage, error) {
    bs := &pprovisioning.BlockStorage{}
    if err := a.dataStore.Get(name, bs); err != nil {
        log.Printf("[WARNING] Failed to get data from db: err='%s'", err.Error())
        return nil, grpcutil.WrapGrpcErrorf(codes.Internal, "Failed to get '%s' from
↳db, please retry or contact for the administrator of this cluster", name)
    } else if bs.Name == "" {
        return nil, grpcutil.WrapGrpcErrorf(codes.NotFound, "")
    }

    if bs.State == pprovisioning.BlockStorage_PENDING {
        return nil, grpcutil.WrapGrpcErrorf(codes.FailedPrecondition, "BlockStorage '
↳%s' is pending", name)
    }

    current := bs.State
    bs.State = pprovisioning.BlockStorage_PENDING
    if err := a.dataStore.Apply(bs.Name, bs); err != nil {
        return nil, grpcutil.WrapGrpcErrorf(codes.Internal, "Failed to apply data for
↳db: err='%s'", err.Error())
    }
    bs.State = current
    tx.PushRollback("free optimistic lock", func() error {
        return a.dataStore.Apply(bs.Name, bs)
    })

    return bs, nil
}

```

Consequences

- PENDING
- network
-

Transaction for update process

Status	accepted

Context

- n0stackgRPC
 - ()

Decision

- github.com/n0stack/n0stack/n0proto.go/pkg/transaction
- Transaction push

Consequences

- github.com/n0stack/n0stack/n0proto.go/pkg/transaction
- API

2.5 References

- [n0stack](#)

2.6 Roadmap

2.6.1 2019 Mar.

- []
- [] etcd
- [] n0cli
- []

2.6.2 2019 Apr.

- []
- [] implement n0bff
- [] ssh

2.6.3 2019 May

- [] mysqlk8s

2.7 Tests

The principles about tests on n0stack.

2.7.1 Test size

- according to <https://testing.googleblog.com/2010/12/test-sizes.html>

small

- unit test about logic
- integration test about side effect
- without side effect, for example...
 - persistent data
 - control middleware
- agent agent ...
 - small
 - agent
 -

Goal

- coverage n0core/pkg/api without agent > 70 %
- coverage n0core/pkg/api with agent > 50 %
- coverage n0core/pkg/datastore/memory > 70 %

medium

- integration test about side effect on standalone
- gRPC fuzzing about logic

Goal

- coverage n0core/pkg/api > 70 %
- coverage n0core/pkg/datastore/etcd > 80 %
- coverage n0core/pkg/driver > 60 %

large

- E2E

2.7.2 TODO

- [x]
- [x] API
- [x] Agent
- [x] Agent
- [x] medium -> small
- [] API

2.8 Tools

2.8.1 Circle CI

-

2.8.2 Travis CI

- test-small

2.8.3 Go Report Card

- Go

2.8.4 CODE CLIMATE

- Golang
 - JS
- protobuf

2.8.5 FOSSA

- - etcd zap (logger) Ignore
 - 1