metalsmith Documentation

Release 0.7.0

MetalSmith Developers

Sep 12, 2018

Contents

CHAPTER 1

Overview

This is a simple tool to provision bare metal machines using OpenStack Bare Metal Service (ironic), OpenStack Image Service (glance) and OpenStack Networking Service (neutron).

- License: Apache License, Version 2.0
- Documentation: https://metalsmith.readthedocs.io
- Source: https://git.openstack.org/cgit/openstack/metalsmith
- Bugs: https://storyboard.openstack.org/#!/project/1000

CHAPTER 2

Installation

```
pip install --user metalsmith
```

CHAPTER 3

CLI Usage

Generic usage is as follows:

```
metalsmith --os-cloud <CLOUD NAME> deploy --image <GLANCE IMAGE> \
    --network <NEUTRON NET> --ssh-public-key <PATH TO SSH PUBLIC KEY> \
    --resource-class <RESOURCE CLASS>
```

This is an example suitable for TripleO (replace `compute` with the profile you want to deploy):

```
source ~/stackrc
metalsmith deploy --image overcloud-full --network ctlplane \
    --capability profile=compute --ssh-public-key ~/.ssh/id_rsa.pub \
    --resource-class baremetal
```

To remove the deployed instance:

```
metalsmith --os-cloud <CLOUD NAME> undeploy <NODE UUID>
```

For all possible options see the built-in help:

```
metalsmith --help
```

CHAPTER 4

Contributing

- Pull requests: Gerrit (see developer's guide)
- Bugs and RFEs: StoryBoard (please do NOT report bugs to Github)

CHAPTER 5

Python API

The main entry point to the API is *metalsmith.Provisioner*.

5.1 metalsmith package

5.1.1 Submodules

metalsmith.exceptions module

exception metalsmith.exceptions.**CapabilitiesNotFound**(*message*, *capabilities*)
Bases: *metalsmith.exceptions.ReservationFailed*

Requested capabilities do not match any nodes.

Variables **requested_capabilities** – Requested node's capabilities.

exception metalsmith.exceptions.**CustomPredicateFailed**(*message*, *nodes*)
Bases: *metalsmith.exceptions.ReservationFailed*

Custom predicate yielded no nodes.

Variables **nodes** – List of nodes that were checked.

exception metalsmith.exceptions.**DeploymentFailure**(*message*, *nodes*)
Bases: *metalsmith.exceptions.Error*

One or more nodes have failed the deployment.

Variables **nodes** – List of failed nodes.

exception metalsmith.exceptions.**Error**
Bases: exceptions.Exception

Base class for Metalsmith errors.

exception metalsmith.exceptions.**InvalidImage**
Bases: *metalsmith.exceptions.Error*

Requested image is invalid and cannot be used.

exception metalsmith.exceptions.**InvalidNIC**
Bases: *metalsmith.exceptions.Error*

Requested NIC is invalid and cannot be used.

exception metalsmith.exceptions.**InvalidNode**
Bases: *metalsmith.exceptions.Error*

This node cannot be deployed onto.

exception metalsmith.exceptions.**NoNodesReserved**(*nodes*)
Bases: *metalsmith.exceptions.ReservationFailed*

All nodes are already reserved or failed validation.

Variables **nodes** – List of nodes that were checked.

exception metalsmith.exceptions.**NodesNotFound**(*resource_class*, *conductor_group*)
Bases: *metalsmith.exceptions.ReservationFailed*

Initial nodes lookup returned an empty list.

Variables

- **requested_resource_class** – Requested resource class.
- **requested_conductor_group** – Requested conductor group to pick nodes from.

exception metalsmith.exceptions.**ReservationFailed**
Bases: *metalsmith.exceptions.Error*

Failed to reserve a suitable node.

This is the base class for all reservation failures.

exception metalsmith.exceptions.**TraitsNotFound**(*message*, *traits*)
Bases: *metalsmith.exceptions.ReservationFailed*

Requested traits do not match any nodes.

Variables **requested_traits** – Requested node's traits.

exception metalsmith.exceptions.**UnknownRootDiskSize**
Bases: *metalsmith.exceptions.Error*

Cannot determine the root disk size.

exception metalsmith.exceptions.**ValidationFailed**
Bases: *metalsmith.exceptions.ReservationFailed*

Validation failed for all requested nodes.

metalsmith.sources module

Image sources to use when provisioning nodes.

class metalsmith.sources.**FilePartitionImage**(*location*, *kernel_location*, *ramdisk_location*, *checksum*)
Bases: *metalsmith.sources.FileWholeDiskImage*

A partition image from a local file location.

> **Warning:** The location must be local to the **ironic-conductor** process handling the node, not to metalsmith itself! Since there is no easy way to determine which conductor handles a node, the same file must be available at the same location to all conductors in the same group.

Create a local file source.

Parameters

- **location** – Location of the image, optionally starting with `file://`.
- **kernel_location** – Location of the kernel of the image, optionally starting with `file://`.
- **ramdisk_location** – Location of the ramdisk of the image, optionally starting with `file://`.
- **checksum** – MD5 checksum of the image.

class metalsmith.sources.**FileWholeDiskImage**(*location*, *checksum*)

Bases: metalsmith.sources._Source

A whole-disk image from a local file location.

> **Warning:** The location must be local to the **ironic-conductor** process handling the node, not to metalsmith itself! Since there is no easy way to determine which conductor handles a node, the same file must be available at the same location to all conductors in the same group.

Create a local file source.

Parameters

- **location** – Location of the image, optionally starting with `file://`.
- **checksum** – MD5 checksum of the image.

class metalsmith.sources.**GlanceImage**(*image*)

Bases: metalsmith.sources._Source

Image from the OpenStack Image service.

Create a Glance source.

Parameters **image** – *Image* object, ID or name.

class metalsmith.sources.**HttpPartitionImage**(*url*, *kernel_url*, *ramdisk_url*, *checksum=None*, *checksum_url=None*)

Bases: *metalsmith.sources.HttpWholeDiskImage*

A partition image from an HTTP(s) location.

Create an HTTP source.

Parameters

- **url** – URL of the root disk image.
- **kernel_url** – URL of the kernel image.
- **ramdisk_url** – URL of the initramfs image.
- **checksum** – MD5 checksum of the root disk image. Mutually exclusive with `checksum_url`.

- **checksum_url** – URL of the checksum file for the root disk image. Has to be in the standard format of the `md5sum` tool. Mutually exclusive with `checksum`.

class metalsmith.sources.**HttpWholeDiskImage**(*url*, *checksum=None*, *checksum_url=None*)
Bases: metalsmith.sources._Source

A whole-disk image from HTTP(s) location.

Some deployment methods require a checksum of the image. It has to be provided via `checksum` or `checksum_url`.

Only `checksum_url` (if provided) has to be accessible from the current machine. Other URLs have to be accessible by the Bare Metal service (more specifically, by **ironic-conductor** processes).

Create an HTTP source.

Parameters

- **url** – URL of the image.
- **checksum** – MD5 checksum of the image. Mutually exclusive with `checksum_url`.
- **checksum_url** – URL of the checksum file for the image. Has to be in the standard format of the `md5sum` tool. Mutually exclusive with `checksum`.

metalsmith.version module

metalsmith.version.**version_info = pbr.version.VersionInfo(metalsmith:0.7.0)**
Package version reported by pbr.

5.1.2 Module contents

class metalsmith.**Instance**(*api*, *node*)
Bases: object

Instance status in metalsmith.

hostname
Node's hostname.

ip_addresses()
Returns IP addresses for this instance.

Returns dict mapping network name or ID to a list of IP addresses.

is_deployed
Whether the node is deployed.

is_healthy
Whether the node is not at fault or maintenance.

nics()
List NICs for this instance.

Returns List of *Port* objects with additional `network` fields with full representations of their networks.

node
Underlying *Node* object.

state
Instance state.

deploying deployment is in progress

active node is provisioned

maintenance node is provisioned but is in maintenance mode

error node has a failure

unknown node in unexpected state (maybe unprovisioned or modified by a third party)

to_dict()
Convert instance to a dict.

uuid
Instance UUID (the same as *Node* UUID for metalsmith).

class metalsmith.**InstanceConfig**(*ssh_keys=None*)
Bases: object

Configuration of the target instance.

The information attached to this object will be passed via a configdrive to the instance's first boot script (e.g. cloud-init).

Variables

- **ssh_keys** – List of SSH public keys.
- **users** – Users to add on first boot.

add_user(*name*, *admin=True*, *password_hash=None*, *sudo=False*, ***kwargs*)
Add a user to be created on first boot.

Parameters

- **name** – user name.
- **admin** – whether to add the user to the admin group (wheel).
- **password_hash** – user password hash, if password authentication is expected.
- **sudo** – whether to allow the user sudo without password.
- **kwargs** – other arguments to pass.

build_configdrive_directory(***kwds*)
Build a configdrive from the provided information.

Parameters

- **node** – *Node* object.
- **hostname** – instance hostname.

Returns a context manager yielding a directory with files

class metalsmith.**Provisioner**(*session=None*, *cloud_region=None*, *dry_run=False*)
Bases: object

API to deploy/undeploy nodes with OpenStack.

Parameters

- **session** – *Session* object (from keystoneauth) to use when making API requests. Mutually exclusive with **cloud_region**.
- **cloud_region** – cloud configuration object (from openstacksdk) to use when making API requests. Mutually exclusive with **session**.

- **dry_run** – boolean value, set to `True` to prevent any API calls from being actually made.

Variables **connection** – *openstacksdk Connection* object used for accessing OpenStack API during provisioning.

list_instances()
List instances deployed by metalsmith.

Returns list of *metalsmith.Instance* objects.

provision_node(*node*, *image*, *nics=None*, *root_size_gb=None*, *swap_size_mb=None*, *config=None*, *hostname=None*, *netboot=False*, *capabilities=None*, *traits=None*, *wait=None*, *root_disk_size=None*)
Provision the node with the given image.

Example:

```
provisioner.provision_node("compute-1", "centos",
                           nics=[{"network": "private"},
                                 {"network": "external"}],
                           root_size_gb=50,
                           wait=3600)
```

Parameters

- **node** – Node object, UUID or name. Will be reserved first, if not reserved already. Must be in the "available" state with maintenance mode off.
- **image** – Image source - one of *sources*, *Image* name or UUID.
- **nics** – List of virtual NICs to attach to physical ports. Each item is a dict with a key describing the type of the NIC: either a port (`{"port": "<port name or ID>"}`) or a network to create a port on (`{"network": "<network name or ID>"}`).
- **root_size_gb** – The size of the root partition. By default the value of the local_gb property is used.
- **swap_size_mb** – The size of the swap partition. It's an error to specify it for a whole disk image.
- **config** – *metalsmith.InstanceConfig* object with the configuration to pass to the instance.
- **hostname** – Hostname to assign to the instance. Defaults to the node's name or UUID.
- **netboot** – Whether to use networking boot for final instances.
- **capabilities** – Requested capabilities of the node. If present, overwrites the capabilities set by *reserve_node()*. Note that the capabilities are not checked against the ones provided by the node - use *reserve_node()* for that.
- **traits** – Requested traits of the node. If present, overwrites the traits set by *reserve_node()*. Note that the traits are not checked against the ones provided by the node - use *reserve_node()* for that.
- **wait** – How many seconds to wait for the deployment to finish, None to return immediately.
- **root_disk_size** – DEPRECATED, use `root_size_gb`.

Returns *metalsmith.Instance* object with the current status of provisioning. If `wait` is not `None`, provisioning is already finished.

Raises *metalsmith.exceptions.Error*

reserve_node(*resource_class=None*, *conductor_group=None*, *capabilities=None*, *traits=None*, *candidates=None*, *predicate=None*)

Find and reserve a suitable node.

Example:

```
node = provisioner.reserve_node("compute",
                                capabilities={"boot_mode": "uefi"})
```

Parameters

- **resource_class** – Requested resource class. If `None`, a node with any resource class can be chosen.
- **conductor_group** – Conductor group to pick the nodes from. Value `None` means any group, use empty string "" for nodes from the default group.
- **capabilities** – Requested capabilities as a dict.
- **traits** – Requested traits as a list of strings.
- **candidates** – List of nodes (UUIDs, names or *Node* objects) to pick from. The filters (for resource class and capabilities) are still applied to the provided list. The order in which the nodes are considered is retained.
- **predicate** – Custom predicate to run on nodes. A callable that accepts a node and returns `True` if it should be included, `False` otherwise. Any exceptions are propagated to the caller.

Returns reserved *Node* object.

Raises *metalsmith.exceptions.ReservationFailed*

show_instance(*instance_id*)

Show information about instance.

Parameters **instance_id** – hostname, UUID or node name.

Returns *metalsmith.Instance* object.

show_instances(*instances*)

Show information about instance.

More efficient than calling *show_instance()* in a loop, because it caches the node list.

Parameters **instances** – list of hostnames, UUIDs or node names.

Returns list of *metalsmith.Instance* objects in the same order as `instances`.

unprovision_node(*node*, *wait=None*)

Unprovision a previously provisioned node.

Parameters

- **node** – *Node* object, *metalsmith.Instance*, hostname, UUID or node name.
- **wait** – How many seconds to wait for the process to finish, None to return immediately.

Returns the latest *Node* object.

wait_for_provisioning(*nodes*, *timeout=None*, *delay=15*)

Wait for nodes to be provisioned.

Loops until all nodes finish provisioning.

Parameters

- **nodes** – List of nodes (UUID, name, *Node* object or *metalsmith.Instance*).
- **timeout** – How much time (in seconds) to wait for all nodes to finish provisioning. If None (the default), wait forever (more precisely, until the operation times out on server side).
- **delay** – Delay (in seconds) between two provision state checks.

Returns List of updated *metalsmith.Instance* objects if all succeeded.

Raises *metalsmith.exceptions.DeploymentFailure* if the deployment failed or timed out for any nodes.

5.2 metalsmith

CHAPTER 6

Ansible Role

6.1 Metalsmith Deployment

This role deploys instances using **metalsmith** CLI.

6.1.1 Variables

The only required variable is:

metalsmith_instances list of instances to provision, see *Instance* for instance description.

The following optional variables provide the defaults for *Instance* attributes:

metalsmith_candidates the default for `candidates`.

metalsmith_capabilities the default for `capabilities`.

metalsmith_conductor_group the default for `conductor_group`.

metalsmith_extra_args the default for `extra_args`.

metalsmith_image the default for `image`.

metalsmith_image_checksum the default for `image_checksum`.

metalsmith_image_kernel the default for `image_kernel`.

metalsmith_image_ramdisk the default for `image_ramdisk`.

metalsmith_netboot the default for `netboot`

metalsmith_nics the default for `nics`.

metalsmith_resource_class the default for `resource_class`.

metalsmith_root_size the default for `root_size`.

metalsmith_ssh_public_keys the default for `ssh_public_keys`.

metalsmith_swap_size the default for `swap_size`.

metalsmith_traits the default for `traits`.

metalsmith_user_name the default for `user_name`, the default value is `metalsmith`.

6.1.2 Instance

Each instances has the following attributes:

candidates (defaults to metalsmith_candidates) list of nodes (UUIDs or names) to be considered for deployment.

capabilities (defaults to metalsmith_capabilities) node capabilities to request when scheduling.

conductor_group (defaults to metalsmith_conductor_group) conductor group to pick nodes from.

> **Note:** Currently it's not possible to specify the default group.

extra_args (defaults to metalsmith_extra_args) additional arguments to pass to the `metalsmith` CLI on all calls.

image (defaults to metalsmith_image) UUID, name or HTTP(s) URL of the image to use for deployment. Mandatory.

image_checksum (defaults to metalsmith_image_checksum) MD5 checksum or checksum file URL for an HTTP(s) image.

image_kernel (defaults to metalsmith_image_kernel) URL of the kernel image if and only if the `image` is a URL of a partition image.

image_ramdisk (defaults to metalsmith_image_ramdisk) URL of the ramdisk image if and only if the `image` is a URL of a partition image.

netboot whether to boot the deployed instance from network (PXE, iPXE, etc). The default is to use local boot (requires a bootloader on the image).

nics (defaults to metalsmith_nics) list of virtual NICs to attach to node's physical NICs. Each is an object with exactly one attribute:

network creates a port on the given network, for example:

```
nics:
  - network: private
  - network: ctlplane
```

port uses the provided pre-created port:

```
nics:
  - port: b2254316-7867-4615-9fb7-911b3f38ca2a
```

resource_class (defaults to metalsmith_resource_class) requested node's resource class.

root_size (defaults to metalsmith_root_size) size of the root partition (in GiB), if partition images are used.

> **Note:** Also required for whole-disk images due to how the Bare Metal service currently works.

ssh_public_keys (defaults to metalsmith_ssh_public_keys) list of file names with SSH public keys to put to the node.

swap_size (defaults to metalsmith_swap_size) size of the swap partition (in MiB), if partition images are used (it's an error to set it for a whole disk image).

traits list of traits the node should have.

user_name (defaults to metalsmith_user_name) name of the user to create on the instance via configdrive. Requires cloud-init on the image.

6.1.3 Example

```
---
- hosts: all
  tasks:
    - include_role:
        name: metalsmith_deployment
      vars:
        metalsmith_image: centos7
        metalsmith_nics:
          - network: ctlplane
        metalsmith_ssh_public_keys:
          - /home/user/.ssh/id_rsa.pub
        metalsmith_instances:
          - hostname: compute-0
            resource_class: compute
            root_size: 100
            swap_size: 4096
            capabilities:
              boot_mode: uefi
            traits:
              - CUSTOM_GPU
          - hostname: compute-1
            resource_class: compute
            root_size: 100
            swap_size: 4096
            capabilities:
              boot_mode: uefi
            user_name: heat-admin
          - hostname: compute-2
            resource_class: compute
            candidates:
              - e63650f2-4e7d-40b2-8932-f5b0e54698c7
              - f19d00dd-60e1-46c8-b83c-782b4d291d9e
          - hostname: control-0
            resource_class: control
            capabilities:
              boot_mode: uefi
            nics:
              - network: ctlplane
              - port: 1899af15-149d-47dc-b0dc-a68614eeb5c4
          - hostname: custom-partition-image
            resource_class: custom
            image: https://example.com/images/custom-1.0.root.img
            image_kernel: https://example.com/images/custom-1.0.vmlinuz
            image_ramdisk: https://example.com/images/custom-1.0.initrd
```

(continues on next page)

(continued from previous page)

```
    image_checksum: https://example.com/images/MD5SUMS
  - hostname: custom-whole-disk-image
    resource_class: custom
    image: https://example.com/images/custom-1.0.qcow2
    image_checksum: https://example.com/images/MD5SUMS
```

CHAPTER 7

Indexes

- genindex
- modindex
- search

Python Module Index

m

Index

W