Juju Resources Documentation

Release 0.1

Cory Johns

March 01, 2016

Contents

Juju Resources provides helpers for Juju Charms to load binary resources from external sources, as well as tools for creating mirrors of external resources for network-restricted deployments.

This is intended as a stop-gap until Juju has core support for resources, as well as to prototype what features are needed.

Using Juju Resources in a Charm requires defining a set of resources in a resources.yaml file, and then using the Python API or the command-line to *`fetch`*, *`verify`*, and *`reference`* the resources.

If you are going to deploy one or more Charms that uses Juju Resources in an environment with limited network connectivity, you can also manually create a mirror of the Charms' resources by using the command-line to *`fetch`* them ahead of time, when you have network access, and then *`serve`* the resources to the charms when you deploy them.

CHAPTER 1

Resource Defintions (`resources.yaml`)

Resources are described in a `resources.yaml` file, which can contain the following sections:

- `resources`
- `optional_resources`
- `options`

1.1 `resources` Section

The file **must** contain a `resources` section containing a mapping of resource names to definitions. Definitions should be mappings with one of the following sets of keys:

URL Resources

- `url` URL for the resource
- `hash` Cryptographic hash for the resource (can also be a URL to a hash)
- `hash_type` Algorithm used to generate the hash; e.g., md5, sha512, etc.

PyPI Resources

- `pypi` A Python package requirement spec, such as `jujuresources>=0.2` (can also be a URL directly to a package)
- `hash` Cryptographic hash for the resource (for use with package URLs)
- `hash_type` Algorithm used to generate the hash; e.g., md5, sha512, etc. (for use with package URLs)

Local or Bundled File Resources

- `file` Path to local file (can be relative to `$CHARM_DIR`)
- `hash` Cryptographic hash for the resource
- `hash_type` Algorithm used to generate the hash; e.g., md5, sha512, etc.

1.2 `optional_resources` Section

The file may contain an `optional_resources` section containing a mapping of resource names to defintions. These take the same form as the required resource definitions.

Optional resources will not be fetched or verified by default, and must either be explicitly named, or the `all` option given.

1.3 `options` Section

The file may contain an `options` section, which supports the following options:

- `output_dir` Location for the fetched resources (default: `./resources`)

1.4 Example

An example `resources.yaml` might be:

```
resources:
    charmhelpers:
        pypi: charmhelpers>=0.2.2
    pyaml:
        pypi: pyaml>=3.11
optional_resources:
    java-x86_64:
        url: http://jr-packages.s3-website-us-east-1.amazonaws.com/ibm/x86_64/java.bin
        hash: b377b7cccdd281bc5e4c4071f80e84a3
        hash_type: md5
    java-amd:
        url: http://jr-packages.s3-website-us-east-1.amazonaws.com/ibm/amd.bin
        hash: e0c54b25e2199c3bb325520d9f9d047a
        hash_type: md5
options:
    ouptut_dir: /var/share/resources
```

CHAPTER 2

API Documentation

jujuresources.**fetch**(*which=None*, *mirror_url=None*, *resources_yaml='resources.yaml'*, *force=False*, *reporthook=None*)

Attempt to fetch all resources for a charm.

Parameters

- **which** (*list*) – A name, or a list of one or more resource names, to fetch. If ommitted, all non-optional resources are fetched. You can also pass `jujuresources.ALL` to fetch all optional *and* required resources.
- **mirror_url** (*str*) – Fetch resources from the given mirror.
- **resources_yaml** (*str*) – Location of the yaml file containing the resource descriptions (default: `./resources.yaml`). Can be a local file name or a remote URL.
- **bool** (*force*) – Force re-downloading of valid resources.
- **reporthook** (*func*) – Callback for reporting download progress. Will be called once for each resource, just prior to fetching, and will be passed the resource name.

Returns True or False indicating whether the resources were successfully downloaded.

jujuresources.**verify**(*which=None*, *resources_yaml='resources.yaml'*)

Verify if some or all resources previously fetched with `fetch_resources()`, including validating their cryptographic hash.

Parameters

- **which** (*list*) – A list of one or more resource names to check. If ommitted, all non-optional resources are verified. You can also pass `jujuresources.ALL` to fetch all optional and required resources.
- **resources_yaml** (*str*) – Location of the yaml file containing the resource descriptions (default: `resources.yaml`). Can be a local file name or a remote URL.
- **output_dir** (*str*) – Override `output_dir` option from *resources_yaml* (this is intended for mirroring via the CLI and it is not recommended to be used otherwise)

Returns True if all of the resources are available and valid, otherwise False.

jujuresources.**install**(*which=None*, *mirror_url=None*, *destination=None*, *skip_top_level=False*, *resources_yaml='resources.yaml'*)

Install one or more resources.

The resource(s) will be fetched, if necessary, and different resource types are handled appropriately (e.g., PyPI resources are installed with `pip`, archive file resources are extracted, non-archive file resources are copied, etc).

For PyPI resources, this is roughly equivalent to the following:

```
    pip install `juju-resources resource_spec $resource` -i $mirror_url
```

Parameters

- **`which`** (*`list`*) – A name, or a list of one or more resource names, to fetch. If ommitted, all non-optional resources are installed.
- **`mirror_url`** (*`str`*) – Fetch resources from the given mirror.
- **`destination`** (*`str`*) – Destination to which to extract or copy file resources.
- **`skip_top_level`** (*`bool`*) – When extracting archive file resources, skip all members that are at the top level of the archive and instead extract all nested members directly into `destination`. E.g., an archive containing `foo/bar.txt` and `foo/qux/baz.txt` will be extracted as `destination/bar.txt` and `destination/qux/baz.txt`.
- **`resources_yaml`** (*`str`*) – Location of the yaml file containing the resource descriptions (default: `resources.yaml`). Can be a local file name or a remote URL.

Returns True if all resources were successfully installed.

`jujuresources.`**`resource_path`**(*resource_name*, *resources_yaml='resources.yaml'*)

Get the local path for a named resource that has been fetched.

This may return `None` if the local path cannot be determined (for example, if the resource has not been fetched yet and needs to be resolved). Even if it returns a path, that path is not guaranteed to exist or be valid; you should always confirm that a resource is available using `verify()` or `fetch()` before using it.

Parameters

- **`resource_name`** (*`str`*) – The name of a resource to resolve.
- **`resources_yaml`** (*`str`*) – Location of the yaml file containing the resource descriptions (default: `./resources.yaml`). Can be a local file name or a remote URL.

`jujuresources.`**`resource_spec`**(*resource_name*, *resources_yaml='resources.yaml'*)

Get the spec for a named resource. This would be the URL for URL resources, the Python package spec for PyPI resources, the full path for local file resources, etc.

Parameters

- **`resource_name`** (*`str`*) – The name of a resource to resolve.
- **`resources_yaml`** (*`str`*) – Location of the yaml file containing the resource descriptions (default: `./resources.yaml`). Can be a local file name or a remote URL.

class `jujuresources.`**`ALL`**

Placeholder to select all resources, optional as well as required.

`jujuresources.`**`config_get`**(*option_name*)

Helper to access a Juju config option when charmhelpers is not available.

Parameters **`option_name`** (*`str`*) – Name of the config option to get the value of

`jujuresources.`**`juju_log`**(*message*, *level='DEBUG'*)

Helper to send Juju log messages when charmhelpers is not available.

Parameters

- **`message`** (*`str`*) – Message to log
- **`level`** (*`str`*) – Log level (DEBUG, INFO, ERROR, WARNING, CRITICAL; default: DEBUG)

Command-line Documentation

`jujuresources.cli.`**`fetch`**`(`*opts*`)`

usage: juju-resources fetch [-h] [-v] [-f] [-q] [-a] [-u MIRROR_URL] [-d OUTPUT_DIR] [-r RESOURCES] [resource_names [resource_names ...]]

Create a local mirror of one or more resources.

Parameters

- **`resource_names`** – Names of specific resources to fetch (defaults to all required, or all if –all is given)
- **`--verbose`** (*`-v`*) – Write download error information to stderr
- **`--force`** (*`-f`*) – Force re-download of valid resources
- **`--quiet`** (*`-q`*) – Suppress output and only set the return code
- **`--all`** (*`-a`*) – Include all optional resources as well as required
- **`--mirror-url`** (*`-u`*) – URL at which the resources are mirrored
- **`--output-dir`** (*`-d`*) – Directory to place the fetched resources (default: ./resources/)
- **`--resources`** (*`-r`*) – File or URL containing the YAML resource descriptions (default: ./resources.yaml)

`jujuresources.cli.`**`install`**`(`*opts*`)`

usage: juju-resources install [-h] [-s] [-D DESTINATION] [-q] [-a] [-u MIRROR_URL] [-d OUTPUT_DIR] [-r RESOURCES] [resource_names [resource_names ...]]

Install one or more resources.

Parameters

- **`resource_names`** – Names of specific resources to verify (defaults to all required, or all if –all is given)
- **`--skip-top-level`** (*`-s`*) – Skip top-level members of archives, and extract children directly to destination
- **`--destination`** (*`-D`*) – Destination for archive or file resources to be installed to
- **`--quiet`** (*`-q`*) – Suppress output and only set the return code
- **`--all`** (*`-a`*) – Include all optional resources as well as required
- **`--mirror-url`** (*`-u`*) – URL at which the resources are mirrored
- **`--output-dir`** (*`-d`*) – Directory containing the fetched resources (default: ./resources/)

- **`--resources`** (*-r*) – File or URL containing the YAML resource descriptions (default: ./resources.yaml)

`jujuresources.cli.`**`resource_path`**(*opts*)

usage: juju-resources resource_path [-h] [-d OUTPUT_DIR] [-r RESOURCES] resource_name

Return the full path to a named resource.

Parameters

- **`resource_name`** – Name of a resource
- **`--output-dir`** (*-d*) – Directory containing the fetched resources (default: ./resources/)
- **`--resources`** (*-r*) – File or URL containing the YAML resource descriptions (default: ./resources.yaml)

`jujuresources.cli.`**`resource_spec`**(*opts*)

usage: juju-resources resource_spec [-h] [-d OUTPUT_DIR] [-r RESOURCES] resource_name

Return the spec (URL, package spec, file, etc) for a named resource.

Parameters

- **`resource_name`** – Name of a resource
- **`--output-dir`** (*-d*) – Directory containing the fetched resources (default: ./resources/)
- **`--resources`** (*-r*) – File or URL containing the YAML resource descriptions (default: ./resources.yaml)

`jujuresources.cli.`**`serve`**(*opts*)

usage: juju-resources serve [-h] [-s SSL_CERT] [-p PORT] [-H HOST] [-d OUTPUT_DIR] [-r RESOURCES]

Run a light-weight HTTP server hosting previously mirrored resources

Parameters

- **`--ssl-cert`** (*-s*) – Path to an SSL certificate file (will run without SSL if not given)
- **`--port`** (*-p*) – Port on which to bind the mirror server
- **`--host`** (*-H*) – IP address on which to bind the mirror server
- **`--output-dir`** (*-d*) – Directory containing the fetched resources (default: ./resources/)
- **`--resources`** (*-r*) – File or URL containing the YAML resource descriptions (default: ./resources.yaml)

`jujuresources.cli.`**`verify`**(*opts*)

usage: juju-resources verify [-h] [-q] [-a] [-d OUTPUT_DIR] [-r RESOURCES] [resource_names [resource_names ...]]

Verify that one or more resources were downloaded successfully.

Parameters

- **`resource_names`** – Names of specific resources to verify (defaults to all required, or all if –all is given)
- **`--quiet`** (*-q*) – Suppress output and only set the return code
- **`--all`** (*-a*) – Include all optional resources as well as required
- **`--output-dir`** (*-d*) – Directory containing the fetched resources (default: ./resources/)

- **`--resources`** (*-r*) – File or URL containing the YAML resource descriptions (default: ./resources.yaml)

CHAPTER 4

Indices and tables

- genindex
- modindex
- search

Python Module Index

Index