| **layout** | **title** |
| --- | --- |
| default | Core types cheat sheet |

A quick reference guide for the core Puppet types.

For detailed information about these types, see the related topic Type reference.

Related topics:

* [Type reference](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html)

{:.concept}

**The Trifecta**

Package/file/service: Learn it, live it, love it. Even if this is the only Puppet you know, you can still get a whole lot done.

package { 'openssh-server':

ensure => installed,

}

file { '/etc/ssh/sshd\_config':

source => 'puppet:///modules/sshd/sshd\_config',

owner => 'root',

group => 'root',

mode => '0640',

notify => Service['sshd'], # sshd restarts whenever you edit this file.

require => Package['openssh-server'],

}

service { 'sshd':

ensure => running,

enable => true,

}

{:.concept}

**file**

Manages files, directories, and symlinks.

{:.section}

**Important Attributes**

* [ensure](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#file-attribute-ensure) -- Whether the file should exist, and what it should be. Allowed values:
  + file (a normal file)
  + directory (a directory)
  + link (a symlink)
  + present (anything)
  + absent
* [path](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#file-attribute-path) -- The full path to the file on disk; **defaults to title.**
* [owner](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#file-attribute-owner) -- By name or UID.
* [group](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#file-attribute-group) -- By name or GID.
* [mode](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#file-attribute-mode) -- Must be specified exactly. Does the right thing for directories.

{:.section}

**For Normal Files**

* [source](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#file-attribute-source) -- Where to download contents for the file. Usually a puppet:/// URL.
* [content](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#file-attribute-content) -- The file's desired contents, as a string. Most useful when paired with [templates](https://docs.puppet.com/guides/templating.html), but you can also use the output of the [file function](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/function.html#file).

{:.section}

**For Directories**

* [source](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#file-attribute-source) -- Where to download contents for the directory, when recurse => true.
* [recurse](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#file-attribute-recurse) -- Whether to recursively manage files in the directory.
* [purge](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#file-attribute-purge) -- Whether unmanaged files in the directory should be deleted, when recurse => true.

{:.section}

**For Symlinks**

* [target](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#file-attribute-target) -- The symlink target. (Required when ensure => link.)

{:.section}

**Other Notable Attributes**

* [backup](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#file-attribute-backup)
* [checksum](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#file-attribute-checksum)
* [force](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#file-attribute-force)
* [ignore](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#file-attribute-ignore)
* [links](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#file-attribute-links)
* [recurselimit](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#file-attribute-recurselimit)
* [replace](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#file-attribute-replace)

{:.concept}

**package**

Manages software packages.

{:.section}

**Important Attributes**

* [name](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#package-attribute-name) -- The name of the package, as known to your packaging system; **defaults to title.**
* [ensure](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#package-attribute-ensure) -- Whether the package should be installed, and what version to use. Allowed values:
  + present
  + latest (implies present)
  + any version string (implies present)
  + absent
  + purged (Potentially dangerous. Ensures absent, then zaps configuration files and dependencies, including those that other packages depend on. Provider-dependent.)
* [source](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#package-attribute-source) -- Where to obtain the package, if your system's packaging tools don't use a repository.
* [provider](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#package-attribute-provider) -- Which packaging system to use (e.g. Yum vs. Rubygems), if a system has more than one available.

{:.concept}

**service**

Manages services running on the node. Like with packages, some platforms have better tools than others, so read up.

You can make services restart whenever a file changes, with the subscribe or notify metaparameters. For more info, read the related topic about relationships

Related topics:

* [Relationships](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/lang_relationships.html)

{:.section}

**Important Attributes**

* [name](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#service-attribute-name) -- The name of the service to run; **defaults to title.**
* [ensure](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#service-attribute-ensure) -- The desired status of the service. Allowed values:
  + running (or true)
  + stopped (or false)
* [enable](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#service-attribute-enable) -- Whether the service should start on boot. Doesn't work on all systems.
* [hasrestart](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#service-attribute-hasrestart) -- Whether to use the init script's restart command instead of stop+start. Defaults to false.
* [hasstatus](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#service-attribute-hasstatus) -- Whether to use the init script's status command. Defaults to true.

{:.section}

**Other Notable Attributes**

If a service has a bad init script, you can work around it and manage almost anything using the [status](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#service-attribute-status), [start](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#service-attribute-start), [stop](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#service-attribute-stop), [restart](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#service-attribute-restart), [pattern](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#service-attribute-pattern), and [binary](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#service-attribute-binary) attributes.

{:.concept}

**Hello World**

{:.concept}

**notify**

Logs an arbitrary message, at the notice log level. This appears in the POSIX syslog or Windows Event Log on the Puppet agent node and is also logged in reports.

notify { "This message is getting logged on the agent node.": }

{:.section}

**Important Attributes**

* [message](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#notify-attribute-message) -- **Defaults to title.**

{:.concept}

**Grab bag**

{:.concept}

[**exec**](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#exec)

Executes an arbitrary command on the agent node. When using execs, you must either make sure the command can be safely run multiple times, or specify that it should only run under certain conditions.

{:.section}

**Important Attributes**

* [command](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#exec-attribute-command) -- The command to run; **defaults to title.** If this isn't a fully-qualified path, use the path attribute.
* [path](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#exec-attribute-path) -- Where to look for executables, as a colon-separated list or an array.
* [returns](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#exec-attribute-returns) -- Which exit codes indicate success. Defaults to 0.
* [environment](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#exec-attribute-environment) -- An array of environment variables to set (for example, ['MYVAR=somevalue', 'OTHERVAR=othervalue']).

{:.section}

**Attributes to Limit When a Command Should Run**

* [creates](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#exec-attribute-creates) -- A file to look for before running the command. The command only runs if the file doesn’t exist.
* [refreshonly](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#exec-attribute-refreshonly) -- If true, the command only run if a resource it subscribes to (or a resource which notifies it) has changed.
* [onlyif](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#exec-attribute-onlyif) -- A command or array of commands; if any have a non-zero return value, the command won't run.
* [unless](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#exec-attribute-unless) -- The opposite of onlyif.

{:.section}

**Other Notable Attributes:**

[cwd](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#exec-attribute-cwd), [group](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#exec-attribute-group), [logoutput](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#exec-attribute-logoutput), , [timeout](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#exec-attribute-timeout), [tries](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#exec-attribute-tries), [try\_sleep](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#exec-attribute-try_sleep), [user](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#exec-attribute-user).

{:.concept}

[**cron**](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#cron)

Manages cron jobs. Largely self-explanatory. On Windows, you should use [scheduled\_task](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#scheduledtask) instead.

cron { 'logrotate':

command => "/usr/sbin/logrotate",

user => "root",

hour => 2,

minute => 0,

}

{:.section}

**Important Attributes**

* [command](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#cron-attribute-command) -- The command to execute.
* [ensure](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#cron-attribute-ensure) -- Whether the job should exist.
  + present
  + absent
* [hour](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#cron-attribute-hour), [minute](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#cron-attribute-minute), [month](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#cron-attribute-month), [monthday](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#cron-attribute-monthday), and [weekday](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#cron-attribute-weekday) -- The timing of the cron job.

{:.section}

**Other Notable Attributes**

[environment](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#cron-attribute-environment), [name](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#cron-attribute-name), [special](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#cron-attribute-special), [target](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#cron-attribute-target), [user](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#cron-attribute-user).

{:.concept}

[**user**](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#user)

Manages user accounts; mostly used for system users.

user { "jane":

ensure => present,

uid => '507',

gid => 'admin',

shell => '/bin/zsh',

home => '/home/jane',

managehome => true,

}

{:.section}

**Important Attributes**

* [name](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#user-attribute-name) -- The name of the user; **defaults to title.**
* [ensure](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#user-attribute-ensure) -- Whether the user should exist. Allowed values:
  + present
  + absent
  + role
* [uid](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#user-attribute-uid) -- The user ID. Must be specified numerically; chosen automatically if omitted. Read-only on Windows.
* [gid](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#user-attribute-gid) -- The user’s primary group. Can be specified numerically or by name. (Not used on Windows; use groupsinstead.)
* [groups](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#user-attribute-groups) -- An array of other groups to which the user belongs. (Don't include the group specified as the gid.)
* [home](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#user-attribute-home) -- The user's home directory.
* [managehome](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#user-attribute-managehome) -- Whether to manage the home directory when managing the user; if you don't set this to true, you'll need to create the user's home directory manually.
* [shell](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#user-attribute-shell) -- The user's login shell.

{:.section}

**Other Notable Attributes**

[comment](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#user-attribute-comment), [expiry](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#user-attribute-expiry), [membership](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#user-attribute-membership), [password](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#user-attribute-password), [password\_max\_age](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#user-attribute-password_max_age), [password\_min\_age](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#user-attribute-password_min_age), [purge\_ssh\_keys](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#user-attribute-purge_ssh_keys), [salt](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#user-attribute-salt).

{:.concept}

[**group**](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#group)

Manages groups.

{:.section}

**Important Attributes**

* [name](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#group-attribute-name) -- The name of the group; **defaults to title.**
* [ensure](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#group-attribute-ensure) -- Whether the group should exist. Allowed values:
  + present
  + absent
* [gid](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#group-attribute-gid) -- The group ID; must be specified numerically, and is chosen automatically if omitted. Read-only on Windows.
* [members](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/type.html#group-attribute-members) -- Users and groups that should be members of the group. Only applicable to certain operating systems; see the full type reference for details.

[cheatsheet\_module.md](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/cheatsheet_module.md)

| **layout** | **title** |
| --- | --- |
| default | Module cheat sheet |

A quick reference to Puppet module terms and concepts.

For detailed explanations of Puppet modules, see the related topics about modules.

Related topics:

* [Module fundamentals](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/modules_fundamentals.html): How to use and write Puppet modules.
* [Installing modules](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/modules_installing.html): How to install pre-built modules from the Puppet Forge.
* [Publishing modules](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/modules_publishing.html): How to publish your modules to the Puppet Forge.
* [Using plug-ins](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/plugins_in_modules.html): How to arrange plug-ins (like custom facts and custom resource types) in modules and sync them to agent nodes.
* [Documenting modules](https://github.com/puppetlabs/puppet-docs/blob/master/cheatsheets/modules_documentation.html): A module README template and information on providing directions for your module.

**Example Module: /etc/puppetlabs/puppet/modules/apache**

{:.section}

**manifests**

This directory holds the module's Puppet code.

* Each .pp file should contain one and only one class or defined type.
* Filenames and class/defined type names are related; see the examples below.
* Within a module, the special $module\_name variable always contains the module's name.

**apache/manifests/init.pp**

class apache {

...

}

init.pp is special; it should contain a class (or defined type) with the same name as the module.

**apache/manifests/vhost.pp**

define apache::vhost

($port, $docroot)

{

...

}

Other classes (and defined types) should be named modulename::filename (without the .pp extension).

**apache/manifests/config/ssl.pp**

class apache::config::ssl {

...

}

Subdirectories add intermediate namespaces.

{:.section}

**files**

Nodes can download any files in this directory from Puppet's built-in file server.

* Use the source attribute to download file contents from the server.
* Use puppet:/// URIs to specify which file to fetch.
* Files in this directory are served at puppet:///modules/modulename/ filename.

**apache/files/httpd.conf**

To fetch this file:

file {'/etc/apache2/httpd.conf':

ensure => file,

source => 'puppet:///modules/apache/httpd.conf',

}

**apache/files/extra/ssl**

Puppet's file server can navigate any subdirectories:

file {'/etc/apache2/httpd-ssl.conf':

ensure => file,

source => 'puppet:///modules/apache/extra/ssl',

}

{:.section}

**lib**

This directory holds Ruby plugins, which can add features to Puppet and Facter.

Capache/lib/puppet/type/apache\_setting.rb\*\* A custom type.

**apache/lib/puppet/parser/functions/htpasswd.rb** A custom function.

apache/lib/facter/apache\_confdir.rb\*\* A custom fact.

{:.section}

**templates**

This directory holds ERB templates.

* Use the template function to create a string by rendering a template.
* Use the content attribute to fill file contents with a string.
* Template files are referenced as modulename/filename.erb.

**apache/templates/vhost.erb**

To use this template:

file {'/etc/apache2/sites-enabled/wordpress.conf':

ensure => file,

content => template('apache/vhost.erb'),

}