



Writing Custom Types to Manage Web-Based Applications



TIM CINEL • BUILD WHISPERER • ATLASSIAN • @TIMCINEL

This Talk

OUR SITUATION

WORKED EXAMPLE

DEMO

NEXT STEPS



Our Situation



Atlassian's Build Engineering team uses Puppet to manage many services, including...



Sonatype Nexus

- Repositories
- Security
- Proxy features



Atlassian Bamboo

- Variables
- Global defaults
- Security



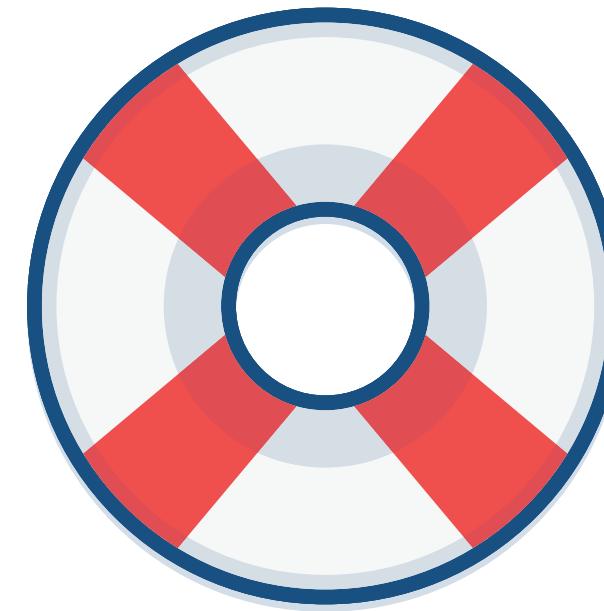
Managing config files has drawbacks...

Uptime



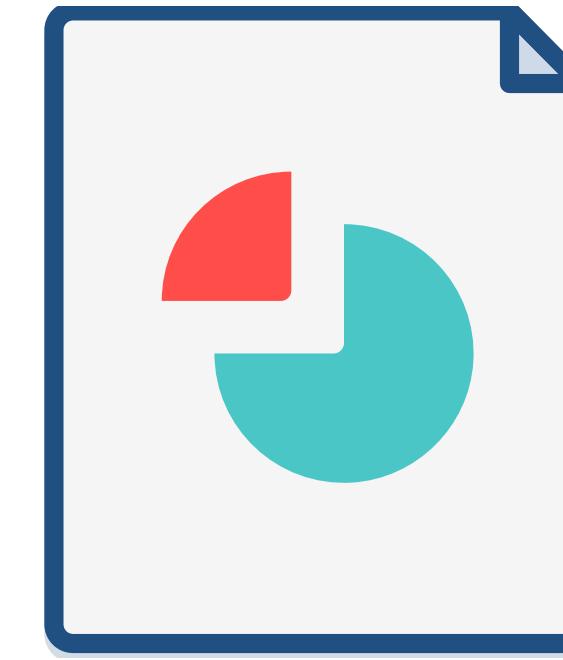
- Nexus's config file changes are picked up after restart
- Changes require brief service outage

Drift



- Changes made using Nexus and Bamboo UI during runtime not detected by Puppet
- Risk of configuration drift

Limited



- Most Bamboo configuration is in its database, not configuration files
- Ability to manage service is limited



How do we address
these drawbacks?

Custom Types.



Concepts Refresher



Concept and terminology refresher

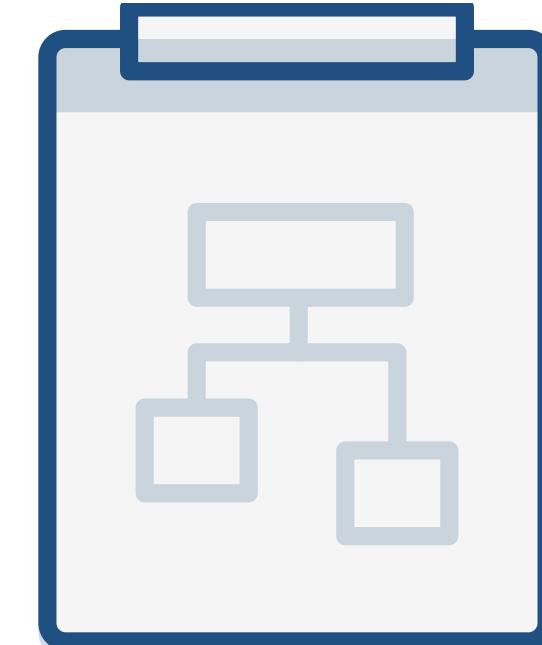
Resource



- Type
- Name
- Attributes

Can be considered as
instances of types

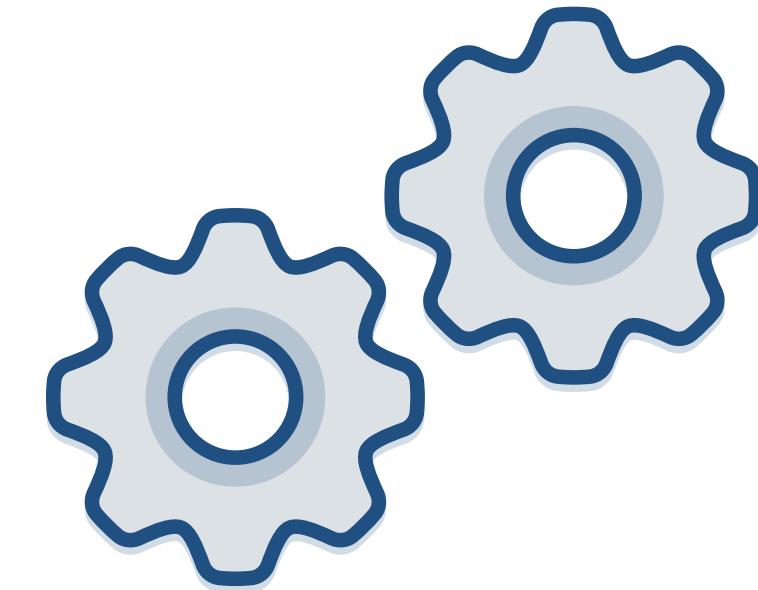
Type



- Native Type
- Ruby implementation
 - Attributes
 - Provider(s)

Allows the representation of a
type of resource in Puppet DSL

Provider



- Ruby implementation
- Type

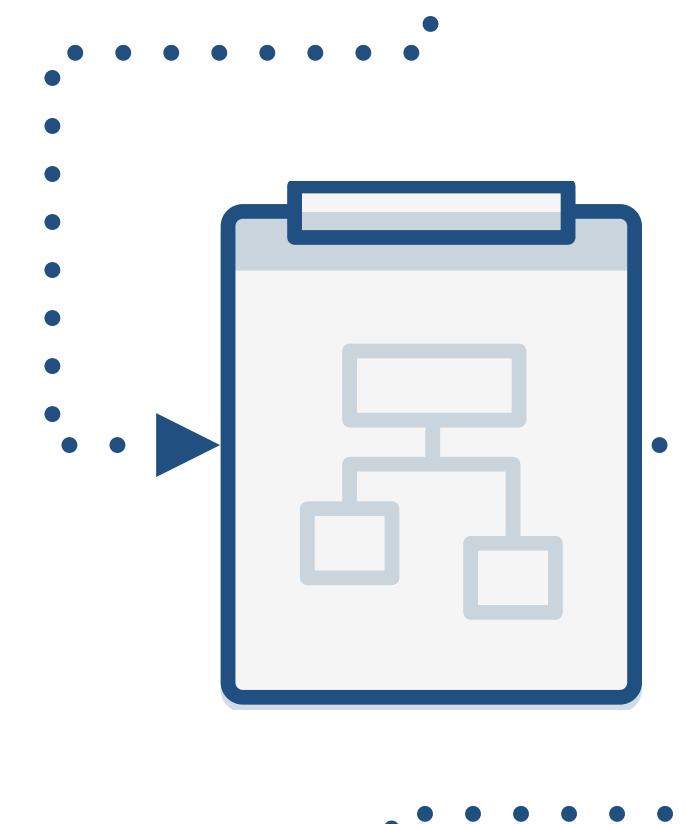
Providers interact with target
system to manage resources



What Are Custom Types?

```
user {'tim':  
  ensure => 'present',  
  uid    => 501,  
  gid    => 501,  
}
```

Manifest



user Type
(Default)

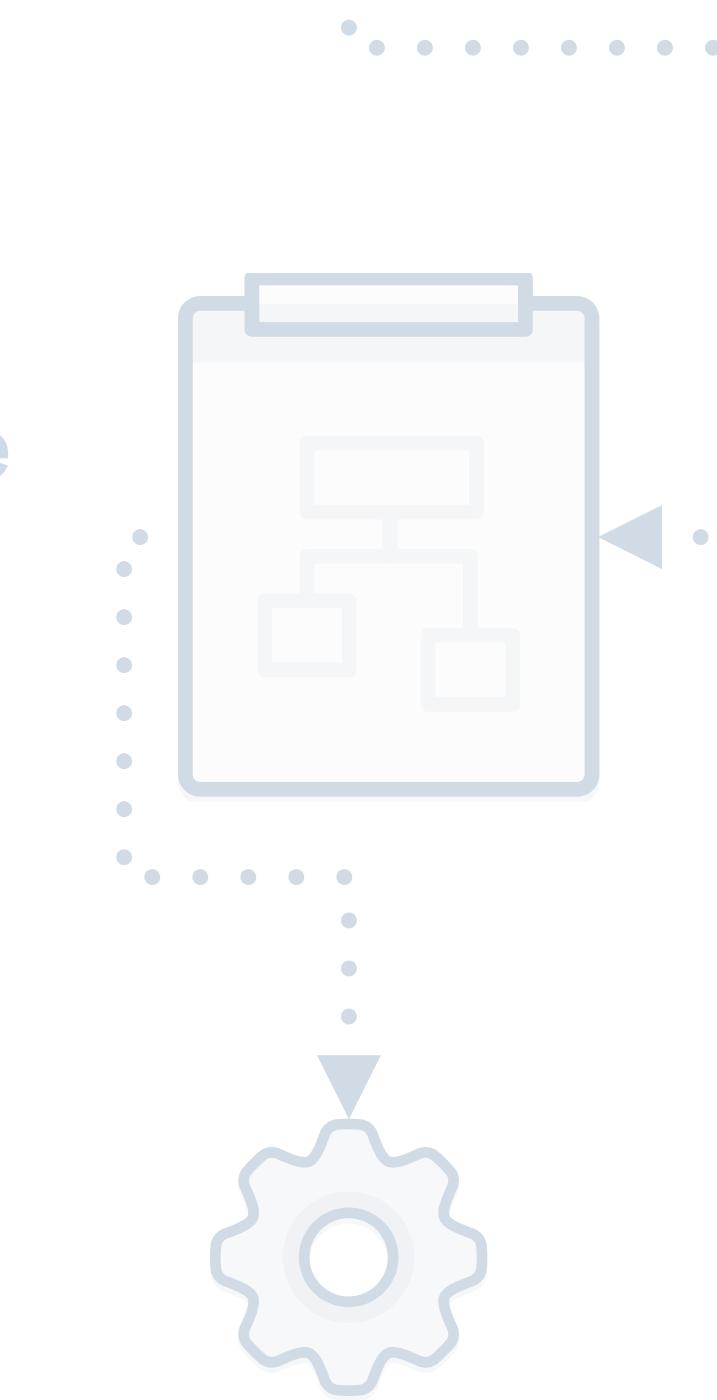
useradd Provider
for user Type

```
$ useradd -g 501 -u 501 tim
```

Manifest

bamboo_agent Type
(Custom)

```
bamboo_agent {'local':  
  enabled => false,  
}
```



rest Provider for
bamboo_agent Type

```
$ curl -X PUT -u "$creds" http://bamboo-server...
```

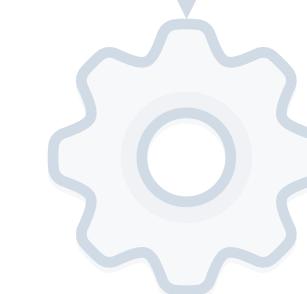
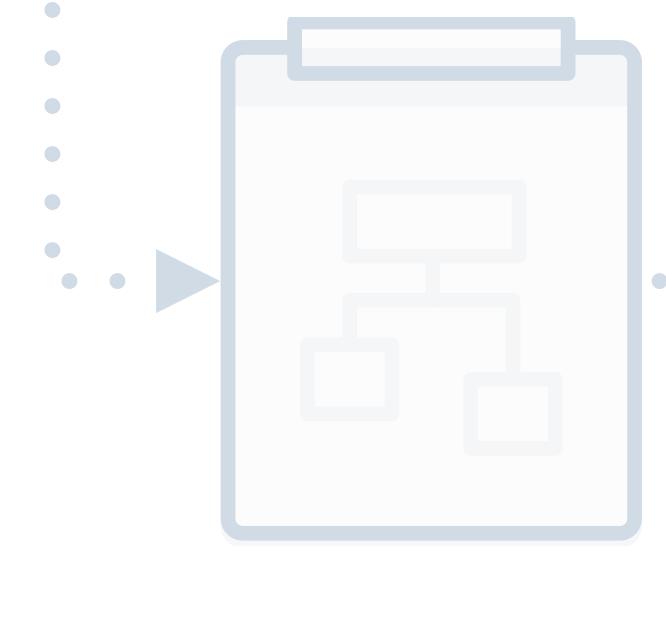


What Are Custom Types?

```
user {'tim':  
  ensure => 'present',  
  uid    => 501,  
  gid    => 501,  
}  
.....
```

Manifest

user Type
(Default)



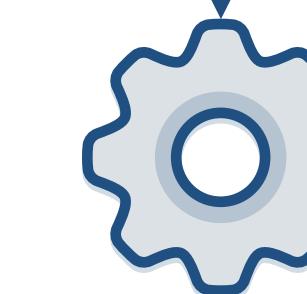
useradd Provider
for user Type

```
$ useradd -g 501 -u 501 tim
```

Manifest

bamboo_agent Type
(Custom)

```
bamboo_agent {'local':  
  enabled => false,  
}  
.....
```



rest Provider for
bamboo_agent Type

```
$ curl -X PUT -u "$creds" http://bamboo-server...
```



Custom Types can be
created to manage
web-based services.



Custom Types

The Good

Fewer Restarts

No need to restart or reload services to pick up configuration changes.

No More Drift

Custom types can directly query the application state runtime.

Fine Configuration

If it's configurable, a custom type can manage it. No need to manually tweak service configuration.



Custom Types

The Bad

Engineering Cost

Somebody has to create and maintain the custom types

More Dependencies

The module containing the custom types, custom providers and libraries

Authentication

Configuration APIs often require credentials



It's not OK to manually configure

20

Bamboo instances



bamboo_rest

https://forge.puppetlabs.com/atlassian/bamboo_rest

Automated away

These properties used to be configured on each server manually.

Not anymore. Thanks Custom Types!

```
24
23 bamboo_audit_log { 'current':
22   audit_log_enabled => true,
21 }
20
19 bamboo_upm_settings { 'current':
18   auto_update_enabled      => false,
17   email_notification_enabled => false,
16   marketplace_connection_enabled => true,
15   request_addon_enabled    => true,
14 }
13
12 bamboo_agent{'local':
11   enabled => false,
10 }
9
8 bamboo_build_expiry { 'current':
7   duration          => 8,
6   period            => 'weeks',
5   builds_to_keep   => 5,
4   cron_expression  => '0 0 10 ? * *',
3   deployments_to_keep => 50,
2   expire_artifacts => true,
1   expire_logs       => true,
60   expire_results   => false,
1   labels_to_exclude => 'retain',
2   maximum_ignored_log_size => -1,
3 }
4
5 bamboo_remote_agent_support { 'current':
6   remote_agents_supported => true;
7 } ->
8 bamboo_elastic_configuration {'current':
```

It's not OK to allow config to drift on

18

Nexus instances



nexus_rest

https://forge.puppetlabs.com/atlassian/nexus_rest

Seamless changes

Changes to these properties used to cause Nexus to restart, resulting in down time.

Not anymore. Thanks Custom Types!

```
23
1 $purge_resources = [
2   'nexus_repository',
3   'nexus_repository_group',
4   'nexus_repository_route',
5   'nexus_repository_target',
6   'nexus_scheduled_task',
7   'nexus_staging_profile',
8   'nexus_staging_ruleset',
9 ]
10
11 resources { $purge_resources:
12   purge => true,
13 }
14
15 nexus_application_server_settings { 'current':
16   baseurl => $baseurl,
17   forceurl => $baseurl_force,
18   timeout => $server_timeout,
19 }
20
21 nexus_connection_settings { 'current':
22   retries => $connection_retries,
23   timeout => $connection_timeout,
24 }
25
26 nexus_system_notification { 'current':
27   enabled => $system_notifications_enabled,
28   emails => $alert_email,
29 }
30
31 nexus_proxy_settings { 'current':
32   http_proxy_enabled => false,
```

Configuration Only

These modules are only responsible for configuring installed services via REST APIs.

Installation of these services is managed by another module.



Worked Example

SET UP

IMPLEMENT A TYPE

WRITE A PROVIDER

ADDING MORE PROPERTIES

INTERACTION BETWEEN TYPES



Worked Example

SET UP

IMPLEMENT A TYPE

WRITE A PROVIDER

ADDING MORE PROPERTIES

INTERACTION BETWEEN TYPES



<https://bitbucket.org/TCinel/puppet-camp-example/>



Helpful Tools

- Puppet
- Ruby
- Bundler
- irb
- curl and Firebug
- Docker



"Tools" (CC BY-NC-ND 2.0) by jlgriffiths

Nexus 3

We're going to create custom types for Nexus 3 in our worked example.

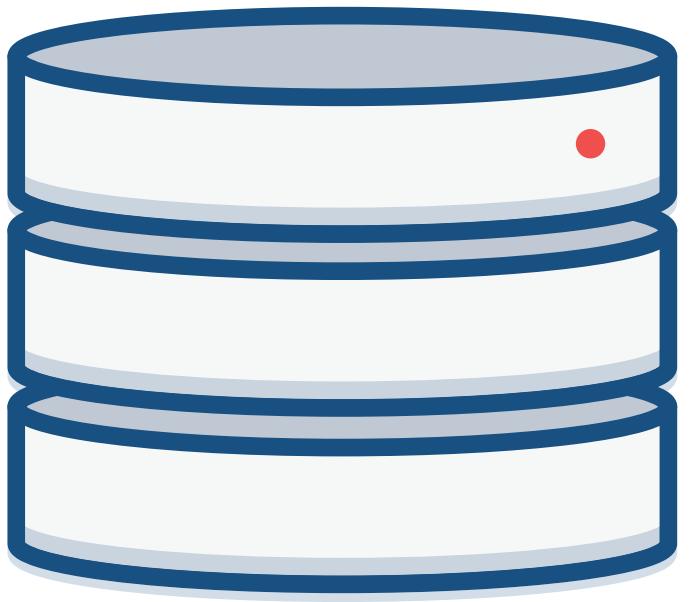
What makes Nexus 3 a good candidate?

- Completely different API to Nexus 2
- Easy to run using Docker
- A repository manager is a good fit with the Puppet resource model



Our Nexus 3 Types

nexus3_repository

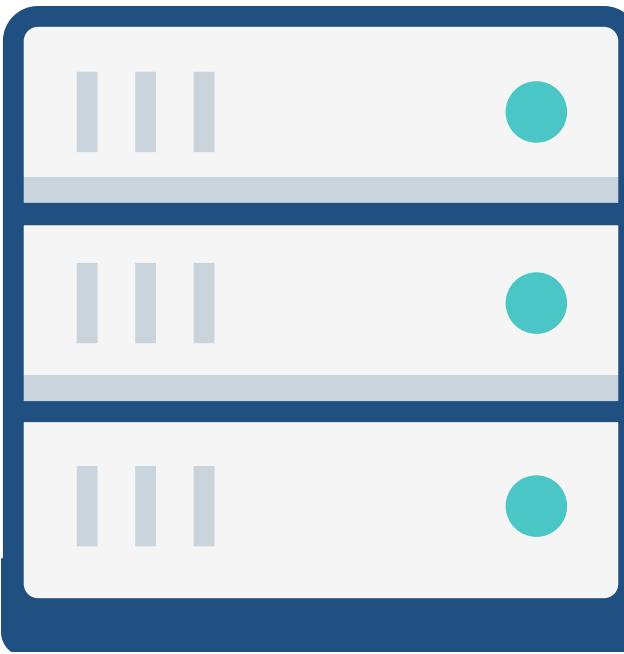


Hosted Maven repository.

Attributes

- name
- online
- blobstore

nexus3_blobstore



Abstract storage used by repositories.

Attributes

- name
- path



Discover Service Endpoints

Useful Endpoints

- Query all resources
- Create new resource
- Delete existing resource
- Update existing resource

Before starting work on a custom type, we should confirm the necessary endpoints are available.



Nexus Endpoint Example

Query Repositories

▼ <http://192.168.99.100:8081/service/extdirect> 192.168.99.100:8081 4.7 KB 192.168.99.100:8081

Headers Post Response JSON Cache

JSON

▶ 0 Object { action="coreui_Blobstore", method="read", type="rpc", more... }
▶ 1 Object { action="coreui_Repository", method="read", type="rpc", more... }
▶ 2 Object { action="coreui_Repository", method="readRecipes", type="rpc", more... }
▶ 3 Object { action="coreui_Repository", method="readReferences", data=[1], more... }

Source

```
[{"action": "coreui_Blobstore", "method": "read", "data": null, "type": "rpc", "tid": 25}, {"action": "coreui_Repository", "method": "read", "data": null, "type": "rpc", "tid": 26}, {"action": "coreui_Repository", "method": "readRecipes", "data": null, "type": "rpc", "tid": 27}, {"action": "coreui_Repository", "method": "readReferences", "data": [{"page": 1, "start": 0, "limit": 25}], "type": "rpc", "tid": 28}]
```



File Structure

- lib/ - All code
 - lib/puppet/ Puppet API code
 - lib/puppet/provider/<type>/<provider>.rb Provider definition
 - lib/puppet/type/<type>.rb Type definition
 - lib/puppet_x helper code

```
" Press ? for help

.. (up a dir)
<sers/tcinel/build/puppet-camp/
lib/
  puppet/
    provider/
      nexus3_blobstore/
        rpc.rb
      nexus3_repository/
        rpc.rb
    type/
      nexus3_blobstore.rb
      nexus3_repository.rb
  puppet_x/
    nexus3/
      client.rb
      extdirect.rb
      nexus_client.rb
      util.rb
  presentation/
  Gemfile
  Gemfile.lock
  TALKNOTES
~
```

Worked Example

SET UP

IMPLEMENT A TYPE

WRITE A PROVIDER

ADDING MORE PROPERTIES

INTERACTION BETWEEN TYPES



Creating a new type

It's as simple as adding these two lines.



The screenshot shows a Mac OS X desktop environment. On the left, a terminal window displays the file structure of a Puppet module:

```
/Users/tcinel/Desktop/nexus3/
└── lib/
    └── puppet/
        └── type/
            └── nexus3_repository.rb
```

On the right, a VIM editor window is open with the file `nexus3_repository.rb`. The code contains three lines of Ruby:

```
1 Puppet::Type.newtype(:nexus3_repository) do
2
3 end
```

A vertical scroll bar is visible on the right side of the VIM window. The status bar at the bottom of the VIM window shows the file name and path: `nexus3_repository.rb (~/Desktop/nexus3/lib/puppet/type) - VIM`.



Creating a new type

It's as simple as adding these two lines.



The screenshot shows a VIM editor window with a file named `nexus3_repository.rb`. The file path is `/Users/tcinel/Desktop/nexus3/lib/puppet/type/nexus3_repository.rb`. The code in the editor is:

```
Puppet::Type.newtype(:nexus3_repository) do
  end
```

The code is highlighted with syntax coloring. The file structure on the left shows a directory tree with `nexus3`, `lib`, `puppet`, and `type` directories, and files `puppet_x`, `Gemfile`, `Gemfile.lock`, and `TALKNOTES`.



Creating a new type

It's as simple as adding these two lines.



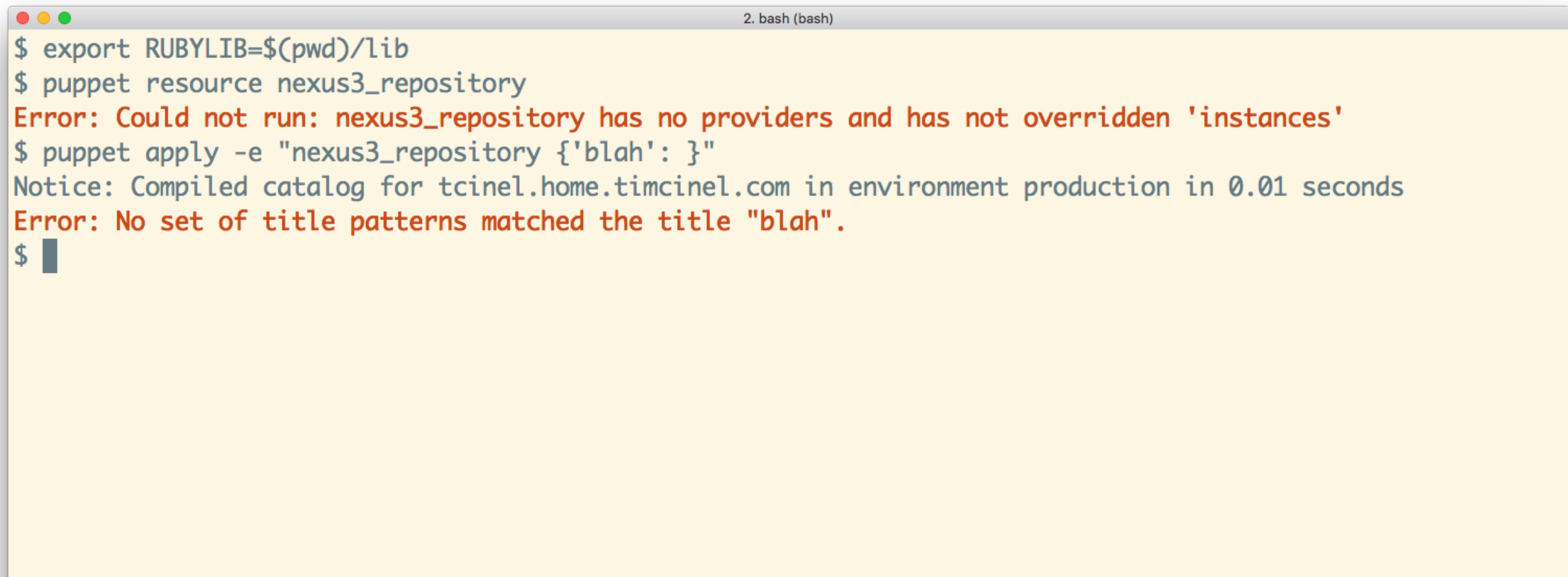
The screenshot shows a terminal window with a file tree on the left and a code editor on the right. The file tree shows a directory structure: `/Users/tcinel/Desktop/nexus3/`, `lib/`, `puppet/`, `type/`, and a file `nexus3_repository.rb`. The file `nexus3_repository.rb` is selected. In the code editor, the following Puppet code is being typed:

```
Puppet::Type.newtype(:nexus3_repository) do
  end
```



It's not quite ready.

At least we have our dev loop now.



```
$ export RUBYLIB=$(pwd)/lib
$ puppet resource nexus3_repository
Error: Could not run: nexus3_repository has no providers and has not overridden 'instances'
$ puppet apply -e "nexus3_repository {'blah': }"
Notice: Compiled catalog for tcinel.home.timcinel.com in environment production in 0.01 seconds
Error: No set of title patterns matched the title "blah".
$
```



It's not quite ready.

At least we have our dev loop now.

**Make sure RUBYLIB contains the path
to lib/ before running puppet**

```
2. bash (bash)
$ export RUBYLIB=$(pwd)/lib
$ puppet resource nexus3_repository
Error: Could not run: nexus3_repository has no providers and has not overridden 'instances'
$ puppet apply -e "nexus3_repository {'blah': }"
Notice: Compiled catalog for tcinel.home.timcinel.com in environment production in 0.01 seconds
Error: No set of title patterns matched the title "blah".
$
```



It's not quite ready.

At least we have our dev loop now.

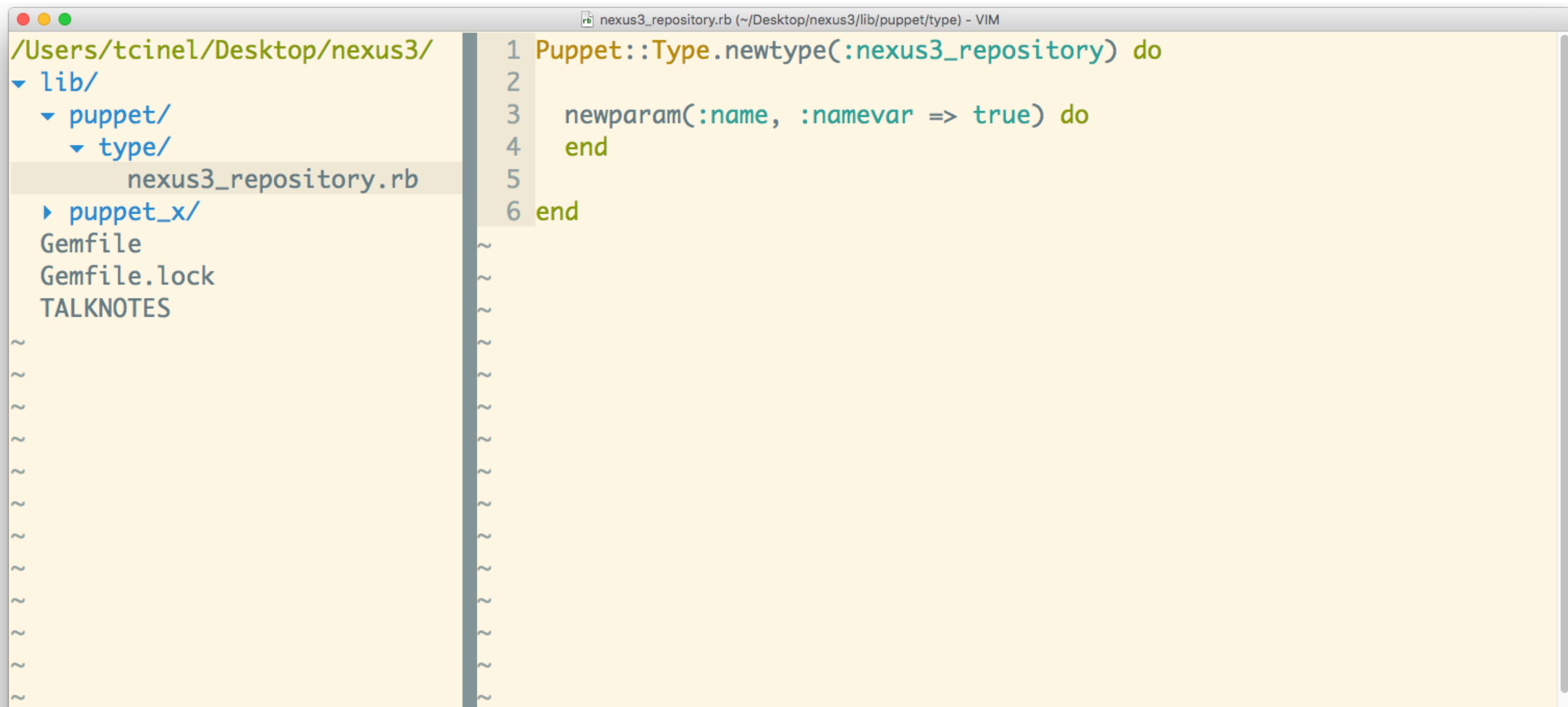
puppet resource and puppet apply are your friends

```
2. bash (bash)
$ export RUBYLIB=$(pwd)/lib
$ puppet resource nexus3_repository
Error: Could not run: nexus3_repository has no providers and has not overridden 'instances'
$ puppet apply -e "nexus3_repository {'blah': }"
Notice: Compiled catalog for tcinel.home.timcinel.com in environment production in 0.01 seconds
Error: No set of title patterns matched the title "blah".
$
```



Add namevar

A type's namevar acts like a primary key



The screenshot shows a VIM editor window with the file `nexus3_repository.rb` open. The file path is `/Users/tcinel/Desktop/nexus3/lib/puppet/type/nexus3_repository.rb`. The code defines a new type:

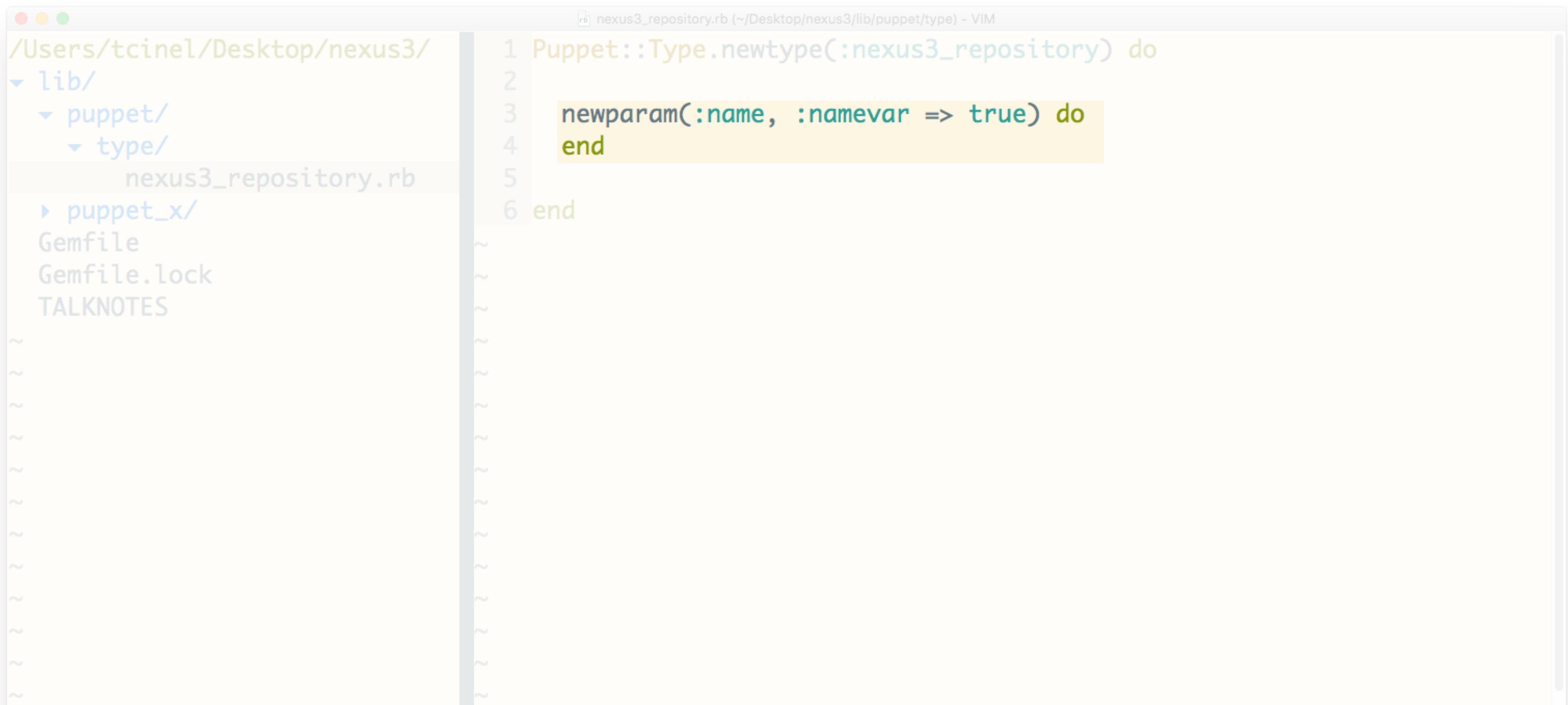
```
Puppet::Type.newtype(:nexus3_repository) do
  newparam(:name, :namevar => true) do
  end
end
```

The file also contains other files in the directory: `puppet_x/`, `Gemfile`, `Gemfile.lock`, and `TALKNOTES`.



Add namevar

A type's namevar acts like a primary key



The screenshot shows a terminal window with a file tree on the left and a Vim editor window on the right.

File tree (left):

- /Users/tcinel/Desktop/nexus3/
- lib/
- puppet/
- type/
- nexus3_repository.rb
- puppet_x/
- Gemfile
- Gemfile.lock
- TALKNOTES

Vim editor (right):

```
rb nexus3_repository.rb (~/Desktop/nexus3/lib/puppet/type) - VIM
1 Puppet::Type.newtype(:nexus3_repository) do
2
3   newparam(:name, :namevar => true) do
4     end
5
6   end
```





2. bash (bash)

```
$ puppet resource nexus3_repository
Error: Could not run: nexus3_repository has no providers and has not overridden 'instances'
$ puppet apply -e "nexus3_repository {'blah': }"
Notice: Compiled catalog for tcinel.home.timcinel.com in environment production in 0.01 seconds
Notice: Finished catalog run in 0.01 seconds
$ █
```



`puppet resource` won't work until we implement a provider

```
2. bash (bash)
$ puppet resource nexus3_repository
Error: Could not run: nexus3_repository has no providers and has not overridden 'instances'
$ puppet apply -e "nexus3_repository {'blah': }"
Notice: Compiled catalog for tcinel.home.timcinel.com in environment production in 0.01 seconds
Notice: Finished catalog run in 0.01 seconds
$
```



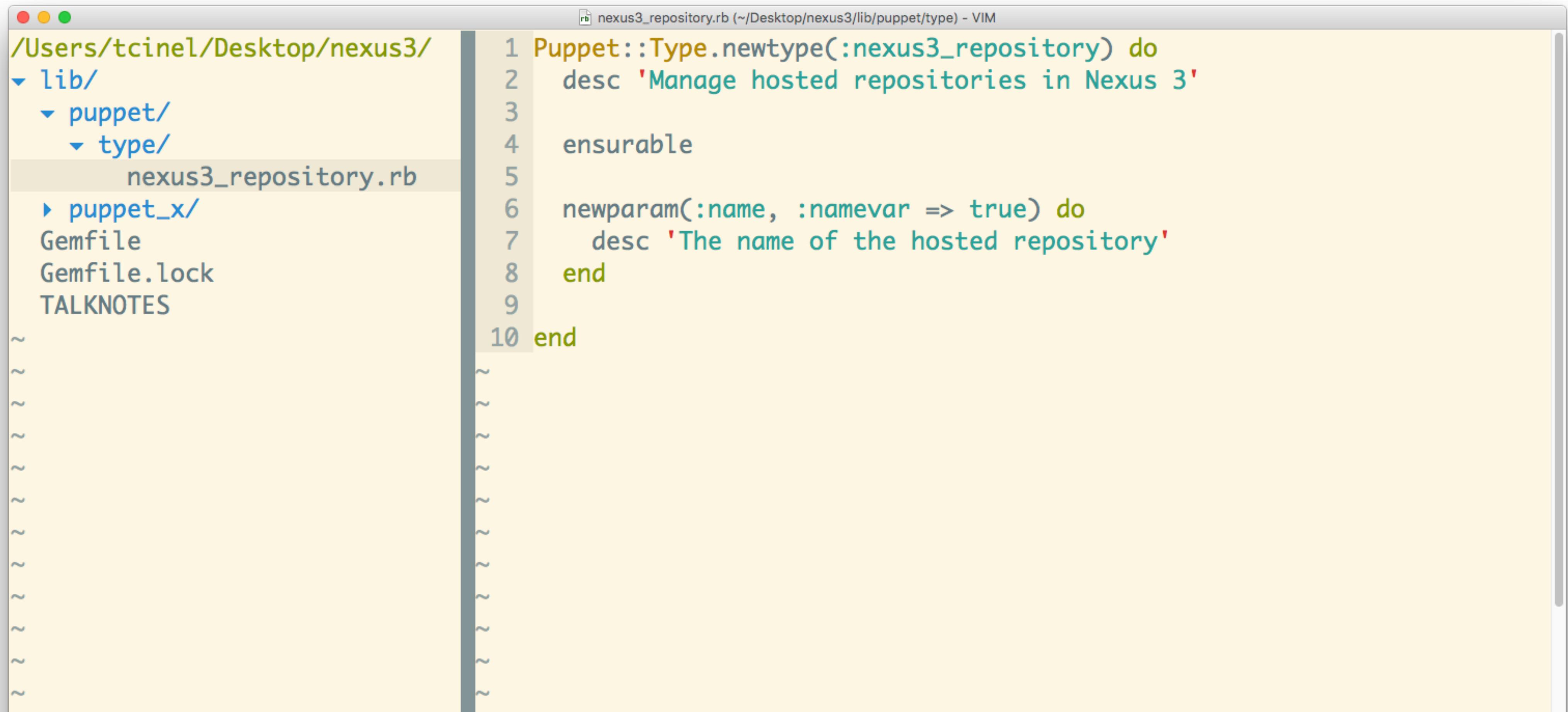
**`puppet apply` works because currently
nexus3_repository is not ensurable**

```
2. bash (bash)
$ puppet resource nexus3_repository
Error: Could not run: nexus3_repository has no providers and has not overridden 'instances'
$ puppet apply -e "nexus3_repository {'blah': }"
Notice: Compiled catalog for tcinel.home.timcinel.com in environment production in 0.01 seconds
Notice: Finished catalog run in 0.01 seconds
$
```



Inline documentation and ensurability

Non-ensurable types (like exec) do not have persistent state.



The image shows a terminal window with a file tree on the left and a Vim editor window on the right. The file tree shows a directory structure under '/Users/tcinel/Desktop/nexus3/'. The Vim window displays the contents of 'nexus3_repository.rb'.

```
rb nexus3_repository.rb (~/Desktop/nexus3/lib/puppet/type) - VIM
1 Puppet::Type.newtype(:nexus3_repository) do
2   desc 'Manage hosted repositories in Nexus 3'
3
4   ensurable
5
6   newparam(:name, :namevar => true) do
7     desc 'The name of the hosted repository'
8   end
9
10 end
```



Inline documentation and ensurability

Non-ensurable types (like exec) do not have persistent state.



The screenshot shows a VIM editor window with two panes. The left pane displays a file tree:

```
/Users/tcinel/Desktop/nexus3/
└── lib/
    └── puppet/
        └── type/
            └── nexus3_repository.rb
```

The right pane shows the contents of the `nexus3_repository.rb` file:

```
rb nexus3_repository.rb (~/Desktop/nexus3/lib/puppet/type) - VIM
1 Puppet::Type.newtype(:nexus3_repository) do
2   desc 'Manage hosted repositories in Nexus 3'
3
4   ensurable
5
6   newparam(:name, :namevar => true) do
7     desc 'The name of the hosted repository'
8   end
9
10 end
```

The word `ensurable` is highlighted with a yellow background. The VIM status bar at the bottom indicates the file is saved.



Inline documentation and ensurability

Adding inline documentation is easy.



The screenshot shows a VIM editor window with a file named `nexus3_repository.rb`. The file is located in a directory structure under `/Users/tcinel/Desktop/nexus3/`, specifically in the `lib/puppet/type` directory. The code defines a new type called `:nexus3_repository` with the following content:

```
Puppet::Type.newtype(:nexus3_repository) do
  desc 'Manage hosted repositories in Nexus 3'
  ensurable
  newparam(:name, :namevar => true) do
    desc 'The name of the hosted repository'
  end
end
```

The code uses Puppet's inline documentation syntax (`desc` and `newparam`) to provide clear descriptions of the type's parameters and its purpose.



Oops. We broke it again.

Looks like we need to implement a provider.

ensurable resources have state, so their provider must have logic to determine state

```
2. bash (bash)
$ puppet apply -e "nexus3_repository {'blah': }"
Notice: Compiled catalog for tcinel.home.timcinel.com in environment production in 0.01 seconds
Error: /Stage[main]/Main/Nexus3_repository[blah]: Could not evaluate: No ability to determine if nexus3_repository exists
Notice: Finished catalog run in 0.01 seconds
$
```



Worked Example

SET UP

IMPLEMENT A TYPE

WRITE A PROVIDER

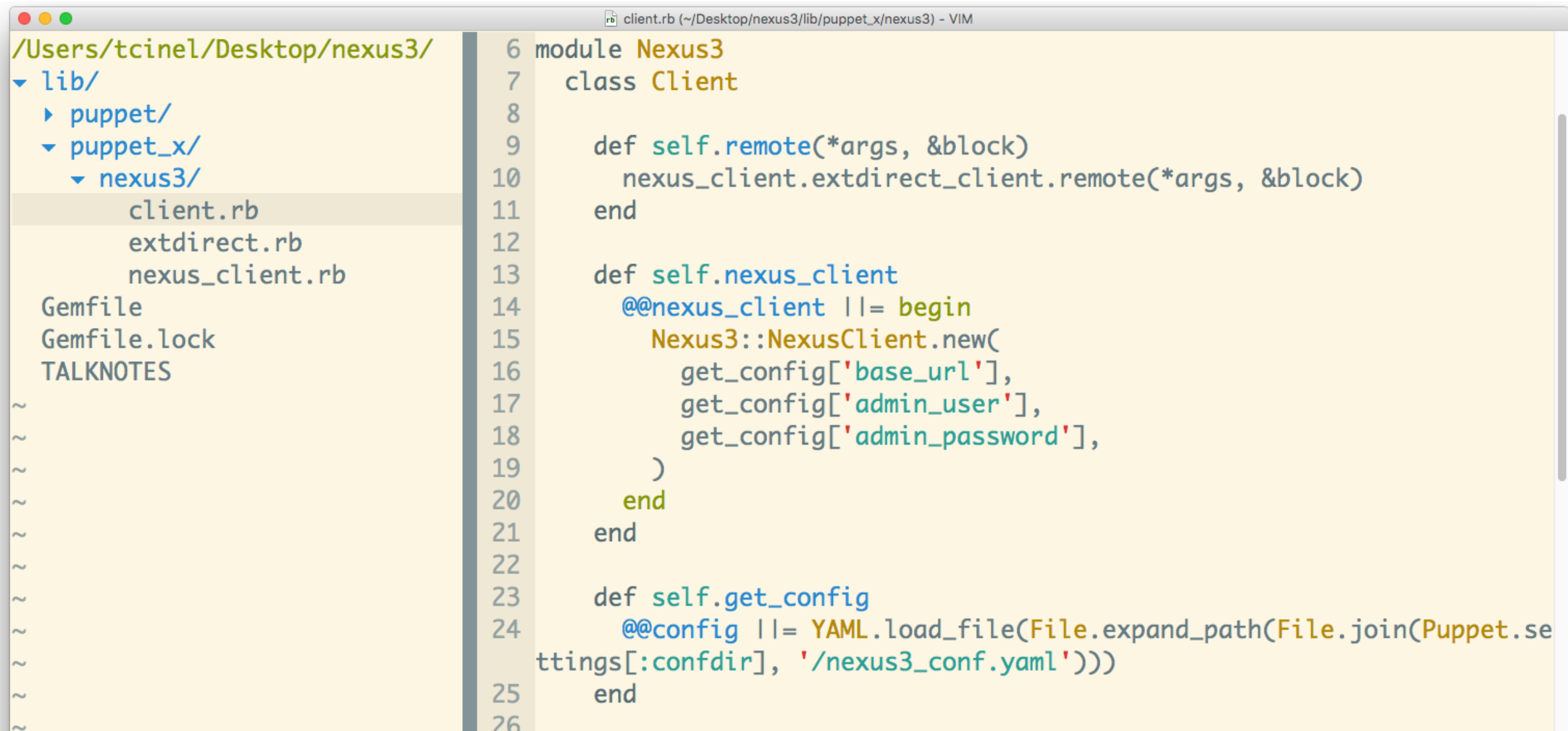
ADDING MORE PROPERTIES

INTERACTION BETWEEN TYPES



Before we start writing the provider...

We could do with some helper classes so we don't get bogged down in implementation details.



The screenshot shows a Vim editor window with the following details:

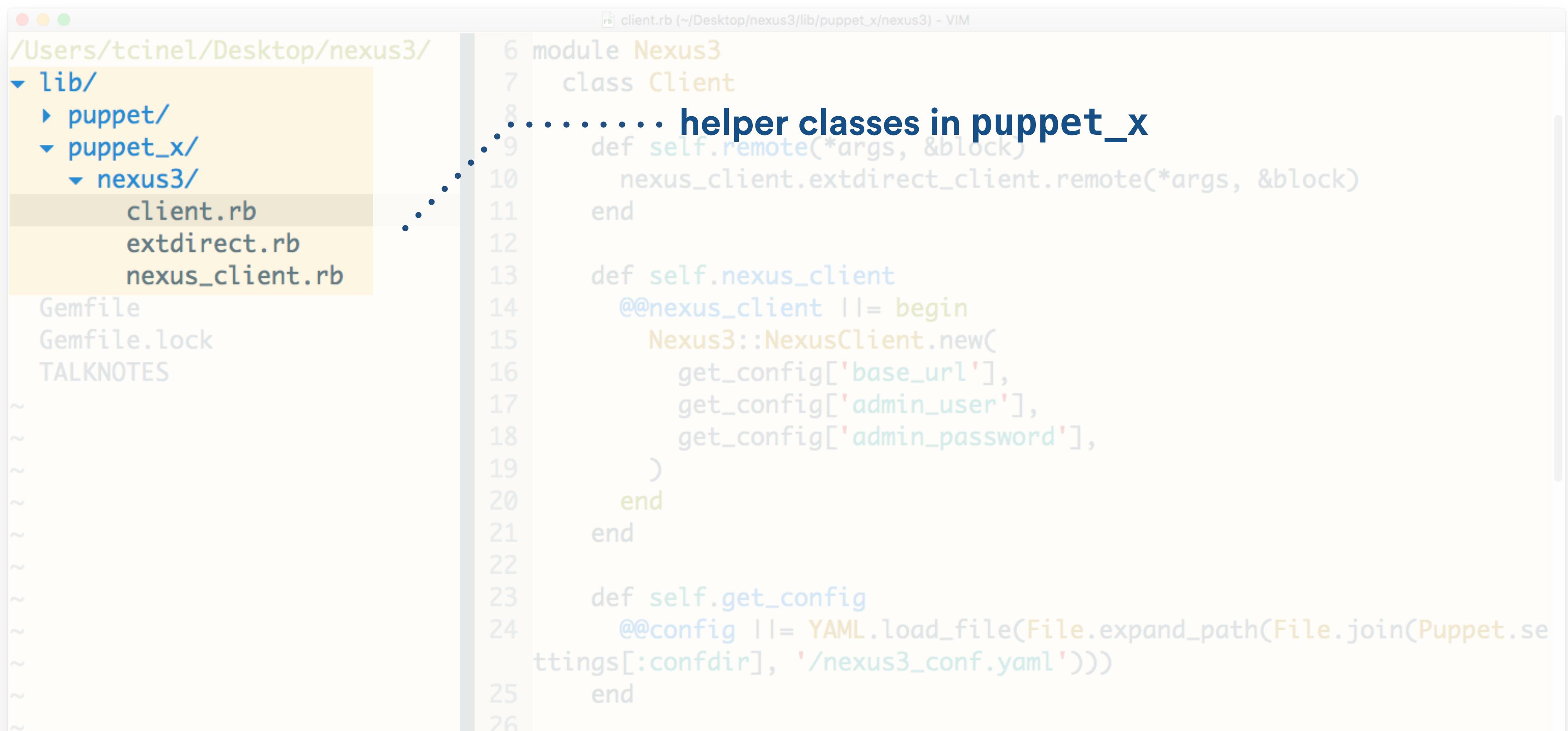
- File Path:** /Users/tcinel/Desktop/nexus3/lib/puppet_x/nexus3/client.rb
- Editor:** Vim (as indicated by the title bar).
- Content:** Ruby code for a Puppet provider named "Client".
- Code Snippet:**

```
6 module Nexus3
7   class Client
8
9     def self.remote(*args, &block)
10      nexus_client.extdirect_client.remote(*args, &block)
11    end
12
13    def self.nexus_client
14      @@nexus_client ||= begin
15        Nexus3::NexusClient.new(
16          get_config['base_url'],
17          get_config['admin_user'],
18          get_config['admin_password'],
19        )
20      end
21    end
22
23    def self.get_config
24      @@config ||= YAML.load_file(File.expand_path(File.join(Puppet.settings[:confdir], '/nexus3_conf.yaml')))
25    end
26  end
```



Before we start writing the provider...

We could do with some helper classes so we don't get bogged down in implementation details.



The screenshot shows a terminal window displaying a file tree and a Vim editor window. The file tree on the left shows a directory structure under '/Users/tcinel/Desktop/nexus3/'. The 'client.rb' file is selected in the tree. The Vim editor window on the right contains Ruby code for a 'Client' class within the 'Nexus3' module. The code includes methods for remote operations and configuration retrieval. A blue box highlights the text 'helper classes in puppet_x'.

```
6 module Nexus3
7   class Client
8     .....
9     def self.remote(*args, &block)
10       nexus_client.extdirect_client.remote(*args, &block)
11     end
12
13     def self.nexus_client
14       @@nexus_client ||= begin
15         Nexus3::NexusClient.new(
16           get_config['base_url'],
17           get_config['admin_user'],
18           get_config['admin_password'],
19         )
20       end
21     end
22
23     def self.get_config
24       @@config ||= YAML.load_file(File.expand_path(File.join(Puppet.se
25 ttings[:confdir], '/nexus3_conf.yaml')))
26     end

```

helper classes in puppet_x



Before we start writing the provider...

We could do with some helper classes so we don't get bogged down in implementation details.



The terminal shows the directory structure:

```
/Users/tcinel/Desktop/nexus3/
└── lib/
    ├── puppet/
    └── puppet_x/
        └── nexus3/
            ├── client.rb
            ├── extdirect.rb
            └── nexus_client.rb

```

A Vim window titled "client.rb (~/Desktop/nexus3/lib/puppet_x/nexus3) - VIM" displays the Ruby code for the "Client" class:

```
6 module Nexus3
7   class Client
8
9     def self.remote(*args, &block)
10      nexus_client.extdirect_client.remote(*args, &block)
11    end
12
13    def self.nexus_client
14      @nexus_client ||= begin
15        Nexus3::NexusClient.new(
16          get_config['base_url'],
17          get_config['admin_user'],
18          get_config['admin_password'],
19        )
20      end
21    end
22
23    def self.get_config
24      @config ||= YAML.load_file(File.expand_path(File.join(Puppet.se
25 ttings[:confdir], '/nexus3_conf.yaml')))
26    end

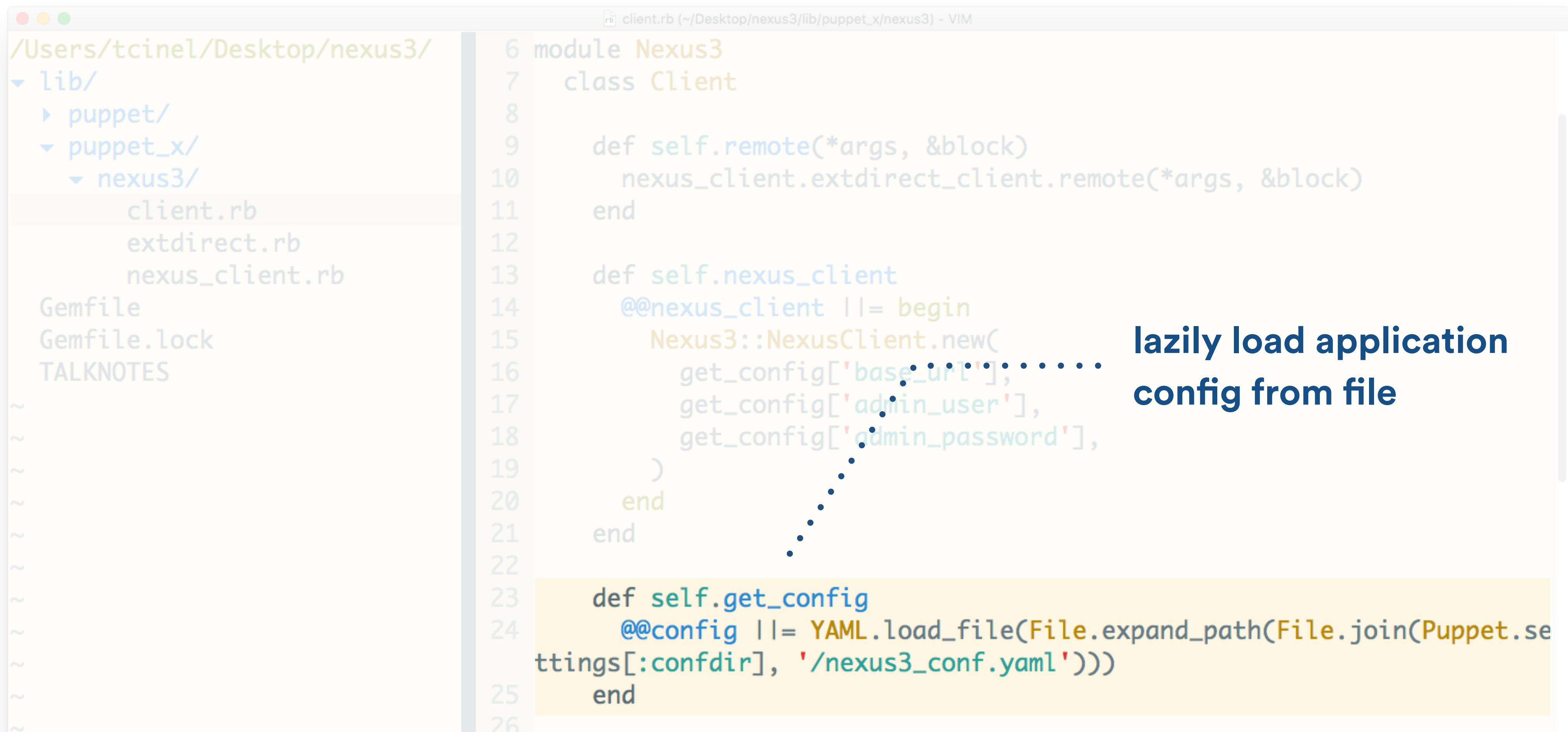
```

A yellow box highlights the line "nexus_client.extdirect_client.remote(*args, &block)". To the right of this box, the text "providers only need to use this class method" is displayed in blue.



Before we start writing the provider...

We could do with some helper classes so we don't get bogged down in implementation details.



The screenshot shows a terminal window with a file tree and a Vim editor window. The file tree on the left shows a directory structure under '/Users/tcinel/Desktop/nexus3/'. The Vim window on the right displays the 'client.rb' file content.

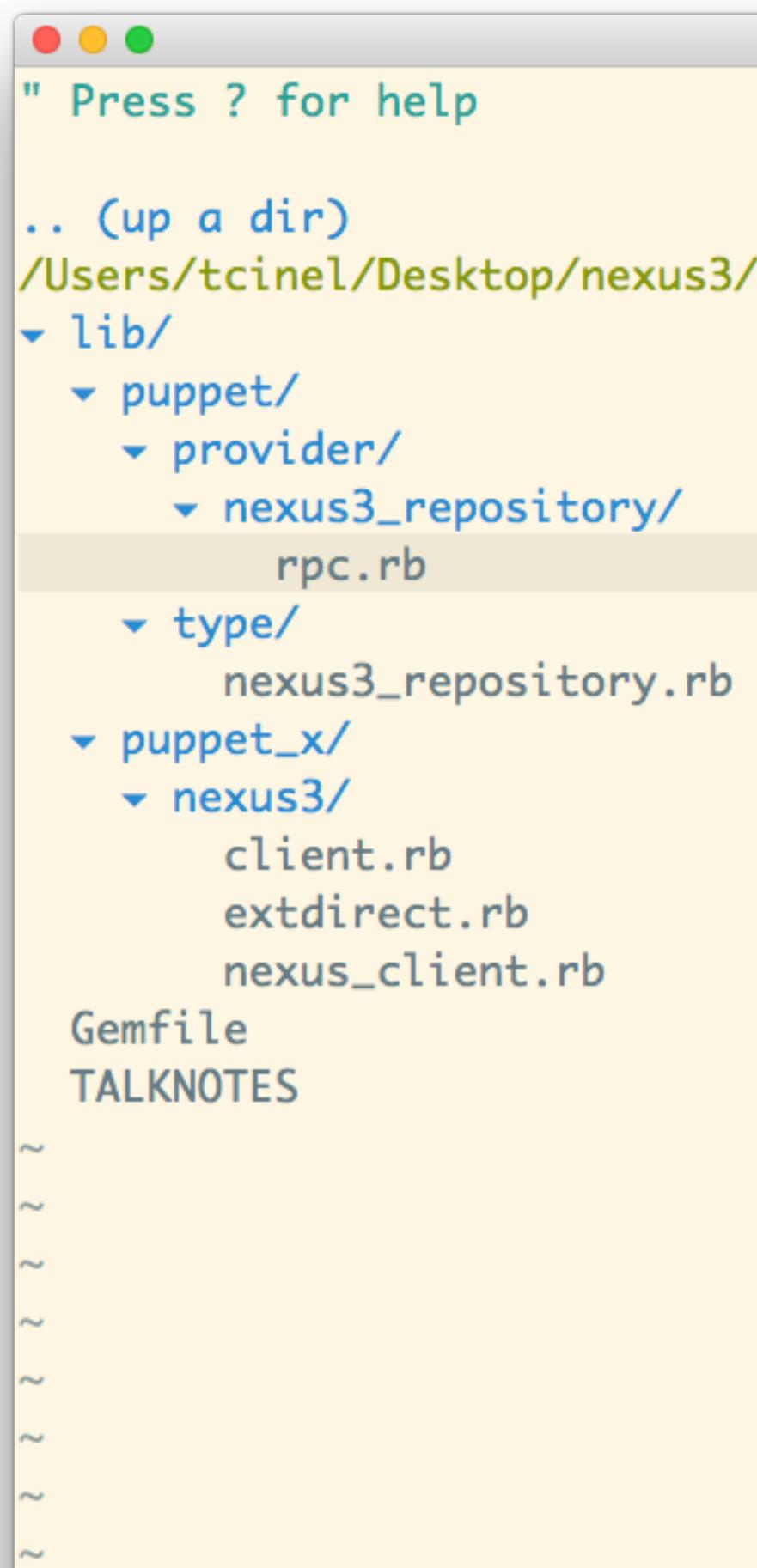
```
6 module Nexus3
7   class Client
8
9     def self.remote(*args, &block)
10      nexus_client.extdirect_client.remote(*args, &block)
11    end
12
13    def self.nexus_client
14      @@nexus_client ||= begin
15        Nexus3::NexusClient.new(
16          get_config['base_url'],
17          get_config['admin_user'],
18          get_config['admin_password'],
19        )
20      end
21    end
22
23    def self.get_config
24      @@config ||= YAML.load_file(File.expand_path(File.join(Puppet.se
25 ttings[:confdir], '/nexus3_conf.yaml')))
26    end

```

lazily load application config from file



A basic provider



The terminal window shows the directory structure of a Puppet provider:

```
" Press ? for help
.. (up a dir)
/Users/tcinel/Desktop/nexus3/
lib/
  puppet/
    provider/
      nexus3_repository/
        rpc.rb
      type/
        nexus3_repository.rb
  puppet_x/
    nexus3/
      client.rb
      extdirect.rb
      nexus_client.rb
Gemfile
TALKNOTES
```

The current file being edited is `rpc.rb`.



```
20 require File.expand_path(File.join(File.dirname(__FILE__), '..', '..', '..', 'puppet_x', 'nexus3',
19
18 Puppet::Type.type(:nexus3_repository).provide(:rpc) do
17
16   def self.instances
15     repositories = Nexus3::Client.remote('coreui_Repository', 'read', nil)['result']['data']
14
13     repositories.select { |repository|
12       repository['recipe'] == 'maven2-hosted'
11     }.collect { |repository|
10       new(
9         :ensure => :present,
8         :name  => repository['name']
7       )
6     }
5   end
4
3   def exists?
2     @property_hash[:ensure] == :present
1   end
21
1 end
```



A basic provider

the provider name is usually the name of the underlying tool used to manage resources



The image shows a terminal window with two panes. The left pane displays the directory structure of a Puppet provider:" Press ? for help
..
../ (up a dir)
/Users/tcinel/Desktop/nexus3/
lib/
 puppet/
 provider/
 nexus3_repository/
 rpc.rb
 type/
 nexus3_repository.rb
 puppet_x/
 nexus3/
 client.rb
 extdirect.rb
 nexus_client.rb
Gemfile
TALKNOTESThe file `rpc.rb` is highlighted. The right pane shows the Ruby code for the `nexus3_repository` provider, specifically the `rpc` provider type:20 require File.expand_path(File.join(File.dirname(__FILE__), '..', '..', 'puppet_x', 'nexus3',
19
18 Puppet::Type.type(:nexus3_repository).provide(:rpc) do
17
16 def self.instances
15 repositories = Nexus3::Client.remote('coreui_Repository', 'read', nil)['result']['data']
14
13 repositories.select { |repository|
12 repository['recipe'] == 'maven2-hosted'
11 }.collect { |repository|
10 new(
9 :ensure => :present,
8 :name => repository['name']
7)
6 }
5 end
4
3 def exists?
2 @property_hash[:ensure] == :present
1 end
21
1 endThe line `Puppet::Type.type(:nexus3_repository).provide(:rpc) do` is highlighted.



A basic provider

if implemented, `self.instances` is used to enumerate all resources of type



The screenshot shows a terminal window with a file tree on the left and the code for the `nexus3_repository` provider on the right.

File Tree:

- " Press ? for help
- .. (up a dir)
- /Users/tcinel/Desktop/nexus3/
- lib/
 - puppet/
 - provider/
 - nexus3_repository/
 - rpc.rb
 - type/
 - nexus3_repository.rb
 - puppet_x/
 - nexus3/
 - client.rb
 - extdirect.rb
 - nexus_client.rb
 - Gemfile
 - TALKNOTES

Code (rpc.rb):

```
20 require File.expand_path(File.join(File.dirname(__FILE__), '..', '..', 'puppet_x', 'nexus3',
19
18 Puppet::Type.type(:nexus3_repository).provide(:rpc) do
17
16   def self.instances
15     repositories = Nexus3::Client.remote('coreui_Repository', 'read', nil)['result']['data']
14
13     repositories.select { |repository|
12       repository['recipe'] == 'maven2-hosted'
11     }.collect { |repository|
10       new(
9         :ensure => :present,
8         :name  => repository['name']
7       )
6     }
5   end
4
3   def exists?
2     @property_hash[:ensure] == :present
1   end
21
1 end
```



A basic provider

exists? provides logic for determining
the presence of a resource



The image shows a terminal window with a file tree and a Vim editor window side-by-side.

File Tree:

- " Press ? for help
- .. (up a dir)
- /Users/tcinel/Desktop/nexus3/
- lib/
 - puppet/
 - provider/
 - nexus3_repository/
 - rpc.rb
 - type/
 - nexus3_repository.rb
 - puppet_x/
 - nexus3/
 - client.rb
 - extdirect.rb
 - nexus_client.rb
 - Gemfile
 - TALKNOTES

VIM Editor: rpc.rb (~/Desktop/nexus3/lib/puppet/provider/nexus3_repository) - VIM

```
20 require File.expand_path(File.join(File.dirname(__FILE__), '...', '...', '...', 'puppet_x', 'nexus3',
19
18 Puppet::Type.type(:nexus3_repository).provide(:rpc) do
17
16   def self.instances
15     repositories = Nexus3::Client.remote('coreui_Repository', 'read', nil)['result']['data']
14
13     repositories.select { |repository|
12       repository['recipe'] == 'maven2-hosted'
11     }.collect { |repository|
10       new(
9         :ensure => :present,
8         :name  => repository['name']
7       )
6     }
5   end
4
3   def exists?
2     @property_hash[:ensure] == :present
1   end
21
1 end
```

A yellow box highlights the `exists?` method definition in the Vim editor.



Aw yeah.

now that we've implemented a basic provider, puppet resource is working



A screenshot of a Mac OS X terminal window titled "2. bash (bash)". The window contains the following text:

```
$ puppet resource nexus3_repository
nexus3_repository { 'maven-release':
  ensure => 'present',
}
nexus3_repository { 'maven-snapshots':
  ensure => 'present',
}
$
```



Aw yeah.

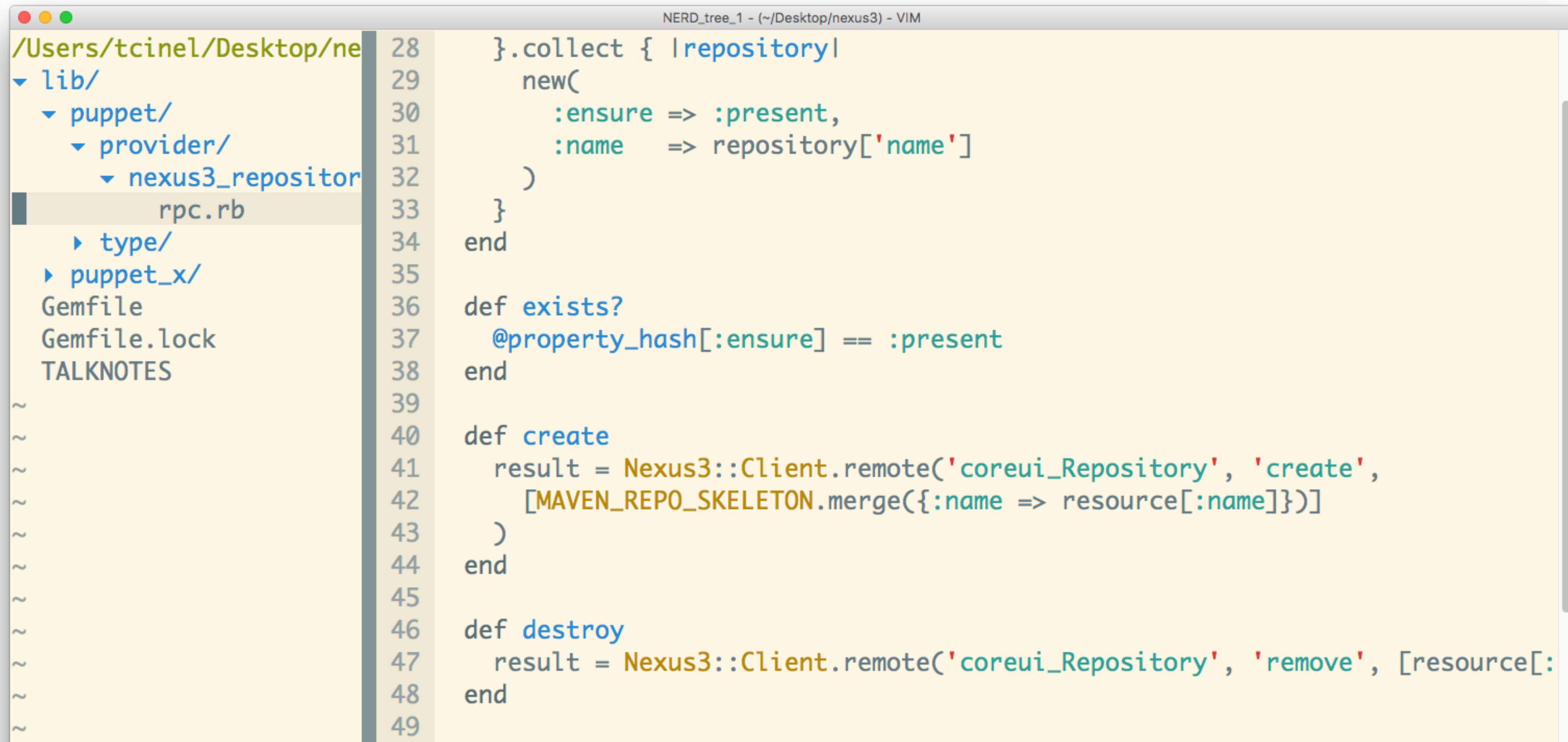
now that we've implemented a basic provider, puppet resource is working

The screenshot shows the Nexus Repository Manager interface. The top navigation bar includes the logo, the text "Nexus Repository Manager oss 3.0.0-03", and a search bar labeled "Search components". Below the header is a sidebar titled "Administration" with the following menu items: "Repository", "Blob Stores", "Repositories" (which is highlighted in blue), and "Content Selectors". To the right of the sidebar is the main content area titled "Repositories" with the sub-instruction "Manage repositories". A "Create repository" button is visible. A table lists two repositories:

	Name ↑	Type	Format
	maven-releases	hosted	maven2
	maven-snapshots	hosted	maven2



Ability to create and destroy



The screenshot shows a Vim window with syntax highlighting for Puppet code. The file path is /Users/tcinel/Desktop/nexus3/lib/puppet/provider/nexus3_repository/rpc.rb. The code defines methods for creating and destroying repositories.

```
28     }.collect { |repository|
29       new(
30         :ensure => :present,
31         :name   => repository['name']
32       )
33     }
34   end
35
36   def exists?
37     @property_hash[:ensure] == :present
38   end
39
40   def create
41     result = Nexus3::Client.remote('coreui_Repository', 'create',
42       [MAVEN_REPO_SKELETON.merge({:name => resource[:name]})]
43     )
44   end
45
46   def destroy
47     result = Nexus3::Client.remote('coreui_Repository', 'remove', [resource[:name]])
48   end
49 }
```



Ability to create and destroy

method used when Puppet determines
that a resource should be created



The screenshot shows a Vim editor window with the title "NERD_tree_1 - (~/Desktop/nexus3) - VIM". The left sidebar displays a file tree:

- /Users/tcinel/Desktop/ne
- lib/
 - puppet/
 - provider/
 - nexus3_repository
 - rpc.rb
 - type/
 - puppet_x/
 - Gemfile
 - Gemfile.lock
 - TALKNOTES

The main pane contains Puppet code:

```
28     }.collect { |repository|
29       new(
30         :ensure => :present,
31         :name   => repository['name']
32       )
33     }
34   end
35
36   def exists?
37     @property_hash[:ensure] == :present
38   end
39
40   def create
41     result = Nexus3::Client.remote('coreui_Repository', 'create',
42       [MAVEN_REPO_SKELETON.merge({:name => resource[:name]})]
43     )
44   end
45
46   def destroy
47     result = Nexus3::Client.remote('coreui_Repository', 'remove', [resource[:name]])
48   end
49
```

A yellow rectangular highlight covers the `create` method definition, which is highlighted in blue. The code uses Puppet's `repository` resource type and `Nexus3::Client` module.



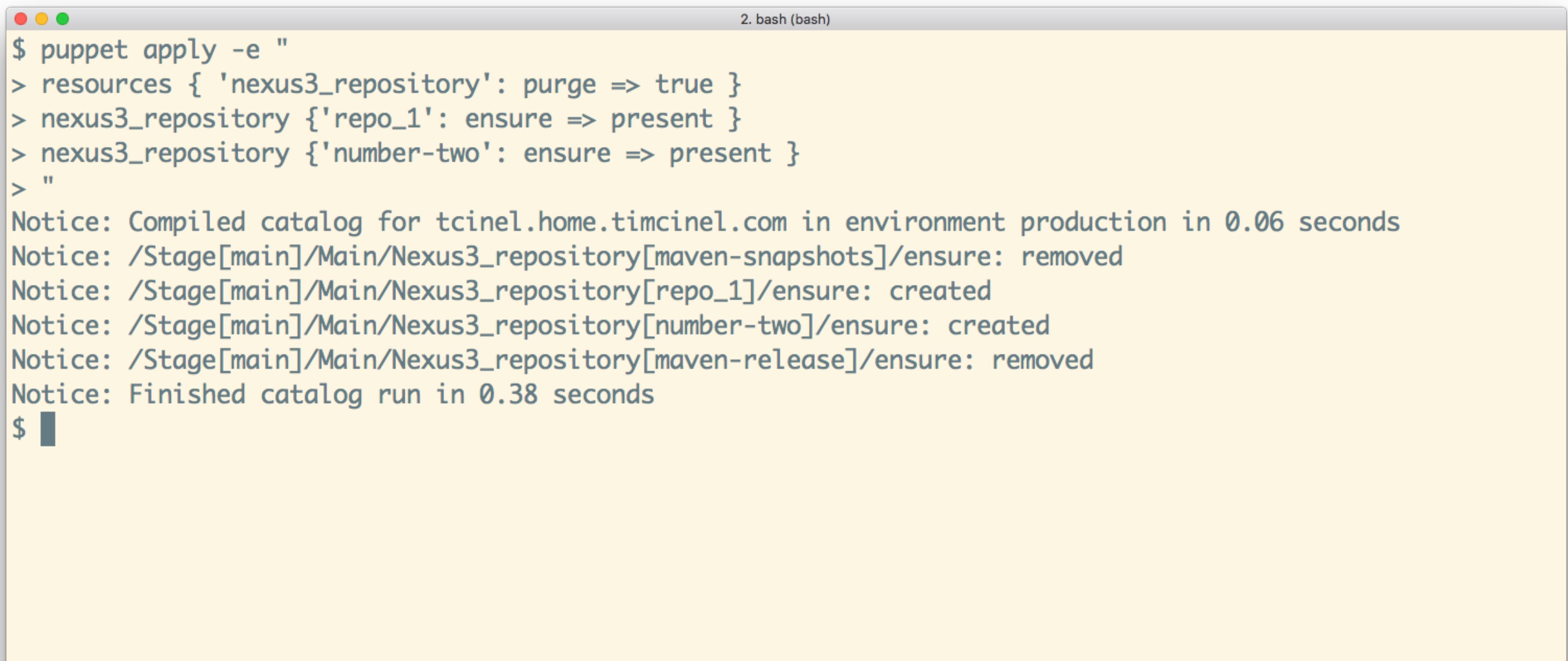
Ability to create and destroy

method used when Puppet wants to
destroy a resource

```
NERD_tree_1 - (~/Desktop/nexus3) - VIM
28   }.collect { |repository|
29     new(
30       :ensure => :present,
31       :name   => repository['name']
32     )
33   }
34 end
35
36 def exists?
37   @property_hash[:ensure] == :present
38 end
39
40 def create
41   result = Nexus3::Client.remote('coreui_Repository', 'create',
42     [MAVEN_REPO_SKELETON.merge({:name => resource[:name]})]
43   )
44 end
45
46 def destroy
47   result = Nexus3::Client.remote('coreui_Repository', 'remove', [resource[:name]])
48 end
49
```



Now we're getting somewhere.



A screenshot of a terminal window titled "2. bash (bash)". The window contains the following text:

```
$ puppet apply -e "
> resources { 'nexus3_repository': purge => true }
> nexus3_repository {'repo_1': ensure => present }
> nexus3_repository {'number-two': ensure => present }
>
Notice: Compiled catalog for tcinel.home.timcinel.com in environment production in 0.06 seconds
Notice: /Stage[main]/Main/Nexus3_repository[maven-snapshots]/ensure: removed
Notice: /Stage[main]/Main/Nexus3_repository[repo_1]/ensure: created
Notice: /Stage[main]/Main/Nexus3_repository[number-two]/ensure: created
Notice: /Stage[main]/Main/Nexus3_repository[maven-release]/ensure: removed
Notice: Finished catalog run in 0.38 seconds
$ "
```



Now we're getting somewhere.

**purges any discovered resources
that are not in the manifest**

```
$ puppet apply -e "
> resources { 'nexus3_repository': purge => true }
> nexus3_repository {'repo_1': ensure => present }
> nexus3_repository {'number-two': ensure => present }
>
Notice: Compiled catalog for tcinel.home.timcinel.com in environment production in 0.06 seconds
Notice: /Stage[main]/Main/Nexus3_repository[maven-snapshots]/ensure: removed
Notice: /Stage[main]/Main/Nexus3_repository[repo_1]/ensure: created
Notice: /Stage[main]/Main/Nexus3_repository[number-two]/ensure: created
Notice: /Stage[main]/Main/Nexus3_repository[maven-release]/ensure: removed
Notice: Finished catalog run in 0.38 seconds
$
```



Now we're getting somewhere.

ensures that repo_1 and number-two are present

```
2. bash (bash)
$ puppet apply -e "
> resources { 'nexus3_repository': purge => true }
> nexus3_repository {'repo_1': ensure => present }
> nexus3_repository {'number-two': ensure => present }
>
Notice: Compiled catalog for tcinel.home.timcinel.com in environment production in 0.06 seconds
Notice: /Stage[main]/Main/Nexus3_repository[maven-snapshots]/ensure: removed
Notice: /Stage[main]/Main/Nexus3_repository[repo_1]/ensure: created
Notice: /Stage[main]/Main/Nexus3_repository[number-two]/ensure: created
Notice: /Stage[main]/Main/Nexus3_repository[maven-release]/ensure: removed
Notice: Finished catalog run in 0.38 seconds
$
```



Now we're getting somewhere.

destroyed unmanaged resources

```
$ puppet apply -e "
> resources { 'nexus3_repository': purge => true }
> nexus3_repository {'repo_1': ensure => present }
> nexus3_repository {'number-two': ensure => present }
>
Notice: Compiled catalog for tcinel.home.timcinel.com in environment production in 0.06 seconds
Notice: /Stage[main]/Main/Nexus3_repository[maven-snapshots]/ensure: removed
Notice: /Stage[main]/Main/Nexus3_repository[repo_1]/ensure: created
Notice: /Stage[main]/Main/Nexus3_repository[number-two]/ensure: created
Notice: /Stage[main]/Main/Nexus3_repository[maven-release]/ensure: removed
Notice: Finished catalog run in 0.38 seconds
$
```



Now we're getting somewhere.

created new resources

```
$ puppet apply -e "
> resources { 'nexus3_repository': purge => true }
> nexus3_repository {'repo_1': ensure => present }
> nexus3_repository {'number-two': ensure => present }
>
Notice: Compiled catalog for tcinel.home.timcinel.com in environment production in 0.06 seconds
Notice: /Stage[main]/Main/Nexus3_repository[maven-snapshots]/ensure: removed
Notice: /Stage[main]/Main/Nexus3_repository[repo_1]/ensure: created
Notice: /Stage[main]/Main/Nexus3_repository[number-two]/ensure: created
Notice: /Stage[main]/Main/Nexus3_repository[maven-release]/ensure: removed
Notice: Finished catalog run in 0.38 seconds
$ "
```



Now we're getting somewhere.

The screenshot shows the Nexus Repository Manager interface. The top bar includes the logo, the text "Nexus Repository Manager", the version "OSS 3.0.0-03", and a search bar. The left sidebar, titled "Administration", has a tree structure with "Repository" expanded, showing "Blob Stores", "Repositories" (which is selected and highlighted in blue), and "Content Selectors". Below this are "Security" and "Privileges". The main right-hand panel is titled "Repositories" and "Manage repositories". It features a "Create repository" button with a plus sign icon. A table lists two repositories: "number-two" (hosted) and "repo_1" (hosted). The table has columns for "Name ↑" and "Type".

	Name ↑	Type
1	number-two	hosted
2	repo_1	hosted



Well, sort of.

**resources are being created again
even though they already exist**

```
2. bash (bash)
$ puppet apply -e "
> resources { 'nexus3_repository': purge => true }
> nexus3_repository {'repo_1': ensure => present }
> nexus3_repository {'number-two': ensure => present }
>
Notice: Compiled catalog for tcinel.home.timcinel.com in environment production in 0.06 seconds
Notice: /Stage[main]/Main/Nexus3_repository[maven-snapshots]/ensure: removed
Notice: /Stage[main]/Main/Nexus3_repository[repo_1]/ensure: created
Notice: /Stage[main]/Main/Nexus3_repository[number-two]/ensure: created
Notice: /Stage[main]/Main/Nexus3_repository[maven-release]/ensure: removed
Notice: Finished catalog run in 0.38 seconds
$ puppet apply -e "
resources { 'nexus3_repository': purge => true }
nexus3_repository {'repo_1': ensure => present }
nexus3_repository {'number-two': ensure => present }
"
Notice: Compiled catalog for tcinel.home.timcinel.com in environment production in 0.05 seconds
Notice: /Stage[main]/Main/Nexus3_repository[repo_1]/ensure: created
Notice: /Stage[main]/Main/Nexus3_repository[number-two]/ensure: created
Notice: Finished catalog run in 0.11 seconds
$
```



Implement prefetch

`prefetch` must be defined if your `exists?` and getter methods don't query the system.



The screenshot shows a VIM editor window with the file `rpc.rb` open. The file is located at `~/Desktop/nexus3/lib/puppet/provider/nexus3_repository`. The code implements the `prefetch` method for a provider. It iterates over resources, finds the provider for each repository, and sets it as the provider for the resource. It also defines the `exists?` method to check if the ensure value is `:present`.

```
rb rpc.rb (~/Desktop/nexus3/lib/puppet/provider/nexus3_repository) - VIM

28     }.collect { |repository|
29         new(
30             :ensure => :present,
31             :name   => repository['name']
32         )
33     }
34 end

36 def self.prefetch(resources)
37     repositories = instances
38     resources.keys.each do |name|
39         if provider = repositories.find { |repository| repository.name == name }
40             resources[name].provider = provider
41         end
42     end
43 end

45 def exists?
46     @property_hash[:ensure] == :present
47 end

48 def create
49     # Implementation
50 end
```



Implement prefetch

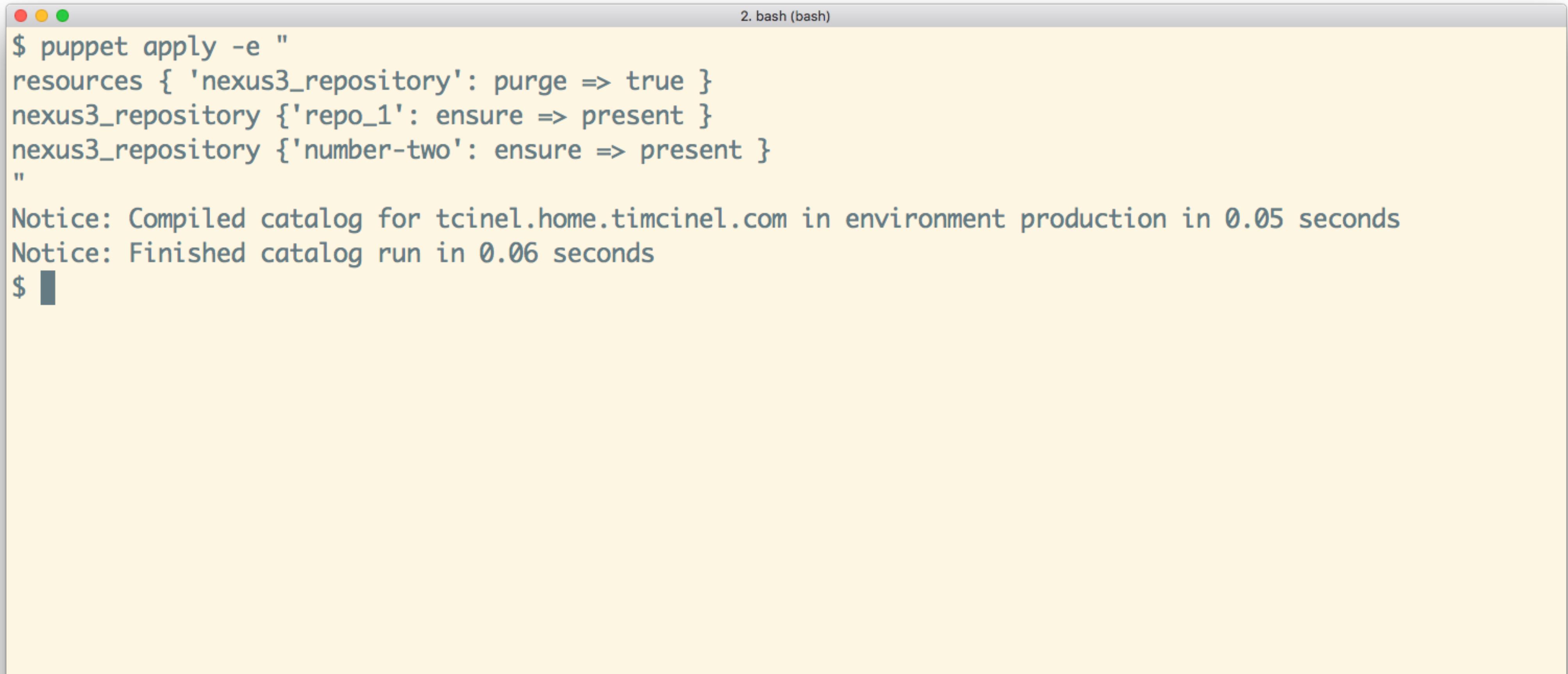
`prefetch` must be defined if your `exists?` and getter methods don't query the system.



```
rb rpc.rb (~/Desktop/nexus3/lib/puppet/provider/nexus3_repository) - VIM
28     }.collect { |repository|
29       new(
30         :ensure => :present,
31         :name   => repository['name']
32       )
33     }
34   end
35
36   def self.prefetch(resources)
37     repositories = instances
38     resources.keys.each do |name|
39       if provider = repositories.find { |repository| repository.name == name }
40         resources[name].provider = provider
41       end
42     end
43   end
44
45   def exists?
46     @property_hash[:ensure] == :present
47   end
48
49   def to_s
50     "Nexus3Repository[#{name}]"
51   end
52 end
```



Yep, now we've got it.



A screenshot of a terminal window titled "2. bash (bash)". The window contains the following text:

```
$ puppet apply -e "
resources { 'nexus3_repository': purge => true }
nexus3_repository {'repo_1': ensure => present }
nexus3_repository {'number-two': ensure => present }
"
Notice: Compiled catalog for tcinel.home.timcinel.com in environment production in 0.05 seconds
Notice: Finished catalog run in 0.06 seconds
$
```



Yep, now we've got it.

no resources recreated

```
$ puppet apply -e "
resources { 'nexus3_repository': purge => true }
nexus3_repository {'repo_1': ensure => present }
nexus3_repository {'number-two': ensure => present }
"
Notice: Compiled catalog for tcinel.home.timcinel.com in environment production in 0.05 seconds
Notice: Finished catalog run in 0.06 seconds
$
```



Worked Example

SET UP

IMPLEMENT A TYPE

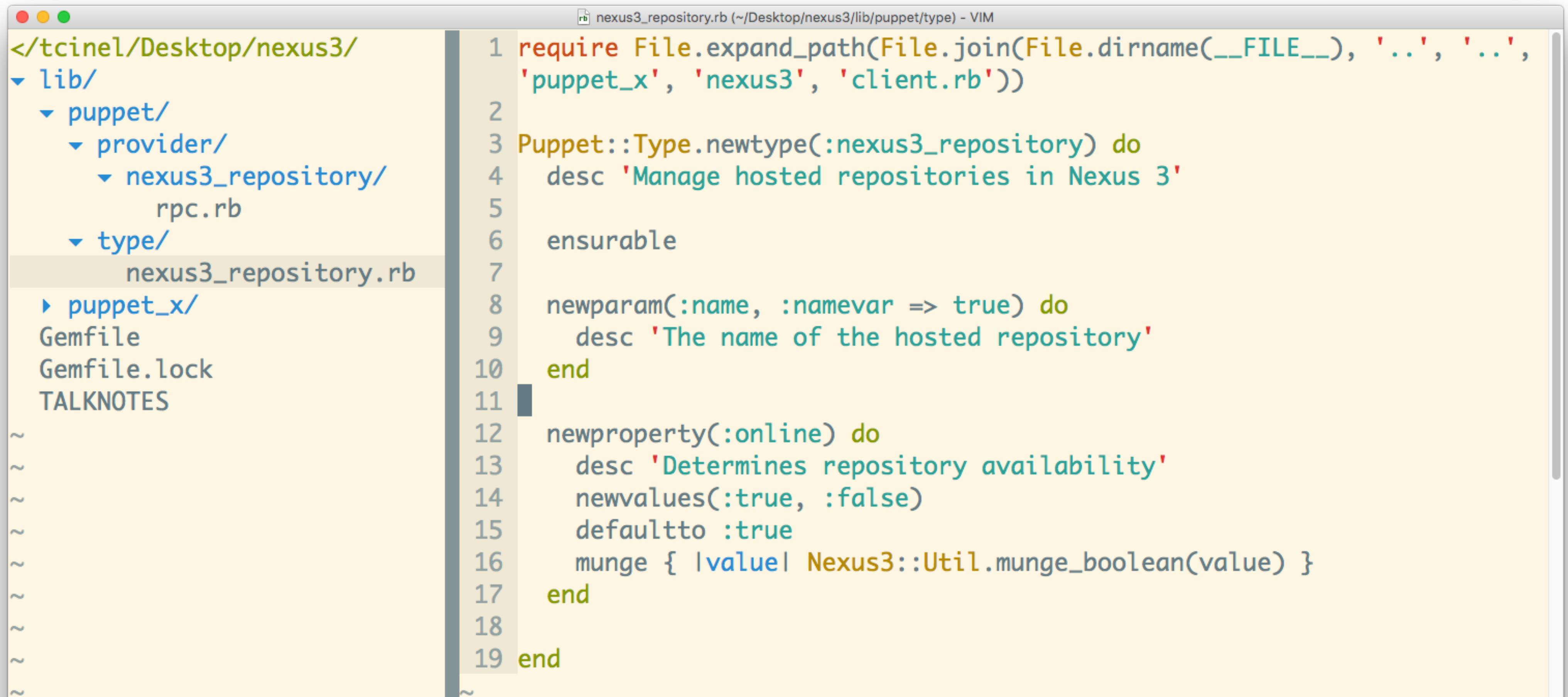
WRITE A PROVIDER

ADDING MORE PROPERTIES

INTERACTION BETWEEN TYPES



Adding an attribute to the Type



The terminal window shows the directory structure:

```
</tcinel/Desktop/nexus3/
└ lib/
  └ puppet/
    └ provider/
      └ nexus3_repository/
        rpc.rb
    └ type/
      nexus3_repository.rb
```

The Vim editor window displays the Ruby code for the `nexus3_repository` type:

```
1 require File.expand_path(File.join(File.dirname(__FILE__), '..', '..',
2                                     'puppet_x', 'nexus3', 'client.rb'))
3 Puppet::Type.newtype(:nexus3_repository) do
4   desc 'Manage hosted repositories in Nexus 3'
5
6   ensurable
7
8   newparam(:name, :namevar => true) do
9     desc 'The name of the hosted repository'
10    end
11
12  newproperty(:online) do
13    desc 'Determines repository availability'
14    newvalues(:true, :false)
15    defaultto :true
16    munge { |value| Nexus3::Util.munge_boolean(value) }
17  end
18
19 end
```



Adding an attribute to the Type

nexus3_repository type now has an online attribute



The image shows a terminal window with a file tree on the left and a Vim editor window on the right.

File Tree:

```
</tcinel/Desktop/nexus3/
└── lib/
    └── puppet/
        └── provider/
            └── nexus3_repository/
                rpc.rb
        └── type/
            nexus3_repository.rb
    └── puppet_x/
    Gemfile
    Gemfile.lock
    TALKNOTES
```

Vim Editor Content: The file is named `nexus3_repository.rb`.

```
1 require File.expand_path(File.join(File.dirname(__FILE__), '..', '..',
2                                     'puppet_x', 'nexus3', 'client.rb'))
3 Puppet::Type.newtype(:nexus3_repository) do
4   desc 'Manage hosted repositories in Nexus 3'
5
6   ensurable
7
8   newparam(:name, :namevar => true) do
9     desc 'The name of the hosted repository'
10    end
11
12  newproperty(:online) do
13    desc 'Determines repository availability'
14    newvalues(:true, :false)
15    defaultto :true
16    munge { |value| Nexus3::Util.munge_boolean(value) }
17  end
18
19 end
```

A yellow rectangular highlight covers the code for the `:online` property definition, starting from line 12 and ending at line 17.



Adding an attribute to the Type

input value restriction



```
</tcinel/Desktop/nexus3/
lib/
  puppet/
    provider/
      nexus3_repository/
        rpc.rb
    type/
      nexus3_repository.rb
  puppet_x/
Gemfile
Gemfile.lock
TALKNOTES
```

nexus3_repository.rb (~/Desktop/nexus3/lib/puppet/type) - VIM

```
1 require File.expand_path(File.join(File.dirname(__FILE__), '..', '..',
'puppet_x', 'nexus3', 'client.rb'))
2
3 Puppet::Type.newtype(:nexus3_repository) do
4   desc 'Manage hosted repositories in Nexus 3'
5
6   ensurable
7
8   newparam(:name, :namevar => true) do
9     desc 'The name of the hosted repository'
10    end
11
12   newproperty(:online) do
13     desc 'Determines repository availability'
14     newvalues(:true, :false)
15     defaultto :true
16     munge { |value| Nexus3::Util.munge_boolean(value) }
17   end
18
19 end
```



Adding an attribute to the Type

input value normalisation



The image shows a terminal window with a file tree on the left and a Vim editor window on the right.

File Tree:

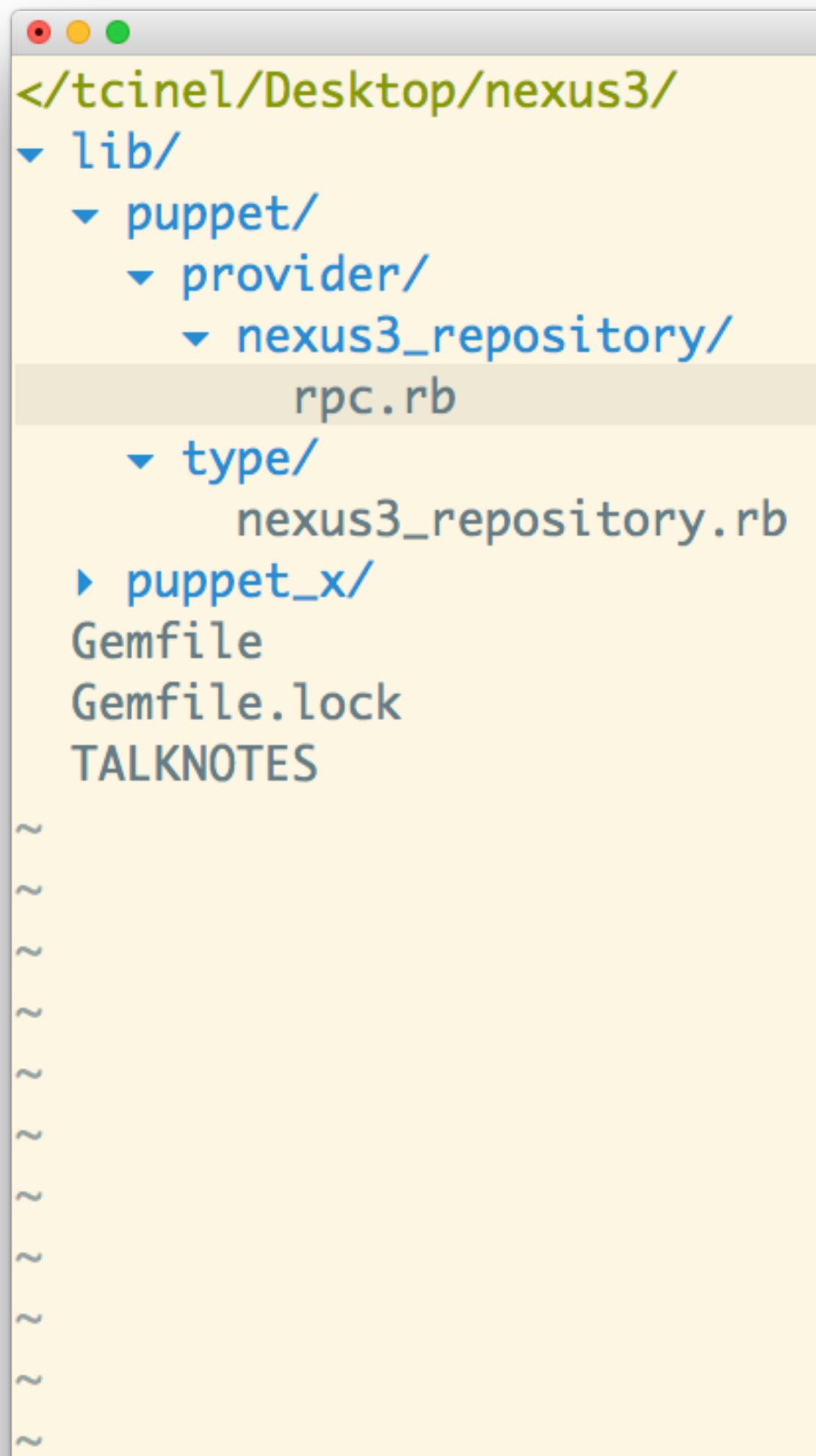
```
</tcinel/Desktop/nexus3/
└── lib/
    ├── puppet/
    │   └── provider/
    │       └── nexus3_repository/
    │           └── rpc.rb
    └── type/
        └── nexus3_repository.rb
```

Vim Editor Content: The file is named `nexus3_repository.rb` and contains Puppet code.

```
1 require File.expand_path(File.join(File.dirname(__FILE__), '../..', 'puppet_x', 'nexus3', 'client.rb'))
2
3 Puppet::Type.newtype(:nexus3_repository) do
4   desc 'Manage hosted repositories in Nexus 3'
5
6   ensurable
7
8   newparam(:name, :namevar => true) do
9     desc 'The name of the hosted repository'
10    end
11
12   newproperty(:online) do
13     desc 'Determines repository availability'
14     newvalues(:true, :false)
15     defaultto :true
16     munge { |value| Nexus3::Util.munge_boolean(value) }
17   end
18
19 end
```



Adding an attribute to the Provider



The screenshot shows a file browser window with the following directory structure:

```
</tcinel/Desktop/nexus3/
  lib/
    puppet/
      provider/
        nexus3_repository/
          rpc.rb
        type/
          nexus3_repository.rb
  puppet_x/
  Gemfile
  Gemfile.lock
  TALKNOTES
```

A file named `rpc.rb` is selected in the file browser.



The code editor displays the `rpc.rb` file content:

```
29 def self.instances
30   repositories = Nexus3::Client.remote('coreui_Repository', 'read', n
31
32   repositories.select { |repository|
33     repository['recipe'] == 'maven2-hosted'
34   }.collect { |repository|
35     new(
36       :ensure => :present,
37       :name   => repository['name'],
38       :online => Nexus3::Util.munge_boolean(repository['online']),
39     )
40   }
41 end
42
43 def map_resource_to_data
44   [
45     MAVEN_REPO_SKELETON.merge(
46       {
47         :name   => resource[:name],
48         :online => Nexus3::Util.sym_to_bool(resource[:online]),
49       }
50     )
51   ]
52 end
```



Adding an attribute to the Provider

fetch the new attribute value from system and normalise it



```
</tcinel/Desktop/nexus3/
└── lib/
    └── puppet/
        └── provider/
            └── nexus3_repository/
                rpc.rb
            └── type/
                nexus3_repository.rb
        └── puppet_x/
Gemfile
Gemfile.lock
TALKNOTES
```

The screenshot shows a file browser window on the left and a code editor window on the right. The file browser displays a directory structure under '/tcinel/Desktop/nexus3/'. The code editor window shows the 'rpc.rb' file from the 'nexus3_repository' provider. The code defines two methods: 'self.instances' and 'map_resource_to_data'. The 'self.instances' method uses the 'Nexus3::Client.remote' method to fetch repositories and then filters them based on their 'recipe' being 'maven2-hosted'. It then collects these repositories into an array of 'new' objects, each with attributes ':ensure' set to ':present', ':name' set to the repository's name, and ':online' set to the result of calling 'Nexus3::Util.munge_boolean' on the repository's 'online' attribute. The 'map_resource_to_data' method is a helper function that takes a resource and merges it with a 'MAVEN_REPO_SKELETON' template, setting the ':name' attribute to the resource's name and the ':online' attribute to the result of calling 'Nexus3::Util.sym_to_bool' on the resource's 'online' attribute.

```
def self.instances
  repositories = Nexus3::Client.remote('coreui_Repository', 'read', n
  repositories.select { |repository|
    repository['recipe'] == 'maven2-hosted'
  }.collect { |repository|
    new(
      :ensure => :present,
      :name   => repository['name'],
      :online => Nexus3::Util.munge_boolean(repository['online']),
    )
  }
end

def map_resource_to_data
  [
    MAVEN_REPO_SKELETON.merge(
      {
        :name   => resource[:name],
        :online => Nexus3::Util.sym_to_bool(resource[:online]),
      }
    )
  ]
end
```



Adding an attribute to the Provider

submit the new attribute to the system on create and update



```
</tcinel/Desktop/nexus3/
└── lib/
    └── puppet/
        └── provider/
            └── nexus3_repository/
                rpc.rb
            └── type/
                nexus3_repository.rb
        └── puppet_x/
Gemfile
Gemfile.lock
TALKNOTES
```

The screenshot shows a file browser window on the left and a code editor window on the right. The file browser displays a directory structure under '/tcinel/Desktop/nexus3/'. The code editor window shows the 'rpc.rb' file from the 'nexus3_repository' provider module. The code defines two methods: 'self.instances' and 'map_resource_to_data'. The 'map_resource_to_data' method includes a call to 'MAVEN_REPO_SKELETON.merge'.

```
def self.instances
  repositories = Nexus3::Client.remote('coreui_Repository', 'read', n
  repositories.select { |repository|
    repository['recipe'] == 'maven2-hosted'
  }.collect { |repository|
    new(
      :ensure => :present,
      :name   => repository['name'],
      :online => Nexus3::Util.munge_boolean(repository['online']),
    )
  }
end

def map_resource_to_data
  [
    MAVEN_REPO_SKELETON.merge(
      {
        :name   => resource[:name],
        :online => Nexus3::Util.sym_to_bool(resource[:online]),
      }
    )
  ]
end
```



Adding an attribute to the Provider



The screenshot shows a file browser window on the left and a code editor window on the right. The file browser displays the directory structure:

```
</tcinel/Desktop/nexus3/
└── lib/
    └── puppet/
        └── provider/
            └── nexus3_repository/
                └── rpc.rb
```

The code editor window shows the content of the `rpc.rb` file:

```
74     Puppet::debug("nexus3_repository destroy result: #{result}")
75 end
76
77 def flush
78   if @flush_required
79     result = Nexus3::Client.remote('coreui_Repository', 'update', map
80
81     Puppet::debug("nexus3_repository update result: #{result}")
82     fail() unless result['result']['success']
83
84     @property_hash = resource.to_hash
85   end
86 end
87
88 def online
89   @property_hash[:online]
90 end
91
92 def online=(value)
93   @flush_required = true
94 end
95
96 end
```



Adding an attribute to the Provider

property getters and setters



The image shows a Vim editor window with two panes. The left pane displays a file tree:

```
</tcinel/Desktop/nexus3/
└── lib/
    └── puppet/
        └── provider/
            └── nexus3_repository/
                rpc.rb
            └── type/
                nexus3_repository.rb
        └── puppet_x/
    Gemfile
    Gemfile.lock
    TALKNOTES
```

The right pane shows the content of the `rpc.rb` file. The code is written in Ruby and includes several Puppet-specific methods and debug statements. A yellow rectangular highlight covers the implementation of the `online` attribute.

```
74     Puppet::debug("nexus3_repository destroy result: #{result}")
75 end
76
77 def flush
78   if @flush_required
79     result = Nexus3::Client.remote('coreui_Repository', 'update', map
80
81   Puppet::debug("nexus3_repository update result: #{result}")
82   fail() unless result['result']['success']
83
84   @property_hash = resource.to_hash
85 end
86 end
87
88 def online
89   @property_hash[:online]
90 end
91
92 def online=(value)
93   @flush_required = true
94 end
95
96 end
```



Adding an attribute to the Provider

flush can be used to implement update logic



The image shows a Vim editor window with two panes. The left pane displays the directory structure of a Puppet provider:

```
</tcinel/Desktop/nexus3/
lib/
  puppet/
    provider/
      nexus3_repository/
        rpc.rb
  type/
    nexus3_repository.rb
puppet_x/
Gemfile
Gemfile.lock
TALKNOTES
```

The right pane shows the content of the `rpc.rb` file. The code implements the `flush` method to handle updates to a Nexus repository. It uses the `Nexus3::Client` to perform a remote update and includes debug logging and failure handling.

```
74   Puppet::debug("nexus3_repository destroy result: #{result}")
75 end
76
77 def flush
78   if @flush_required
79     result = Nexus3::Client.remote('coreui_Repository', 'update', map
80
81     Puppet::debug("nexus3_repository update result: #{result}")
82     fail() unless result['result']['success']
83
84     @property_hash = resource.to_hash
85   end
86 end
87
88 def online
89   @property_hash[:online]
90 end
91
92 def online=(value)
93   @flush_required = true
94 end
95
96 end
```



Aw yeah.



A screenshot of a terminal window titled "2. bash (bash)". The window contains the following Puppet configuration code:

```
$ puppet resource nexus3_repository
nexus3_repository { 'number-two':
  ensure => 'present',
  online => 'true',
}
nexus3_repository { 'repo_1':
  ensure => 'present',
  online => 'true',
}
$ █
```

The code defines two repositories: "number-two" and "repo_1", both set to be present and online. The terminal prompt ends with a blue cursor icon.



Aw yeah.

puppet resource is reporting online status

```
2. bash (bash)
$ puppet resource nexus3_repository
nexus3_repository { 'number-two':
  ensure => 'present',
  online => 'true',
}
nexus3_repository { 'repo_1':
  ensure => 'present',
  online => 'true',
}
$
```



Aw yeah.

puppet apply updated the online attribute

```
$ puppet apply -e "
resources { 'nexus3_repository': purge => true }
nexus3_repository {'repo_1': ensure => present, online => true }
nexus3_repository {'number-two': ensure => present, online => false }
"
Notice: Compiled catalog for tcinel.home.timcinel.com in environment production in 0.07 seconds
Notice: /Stage[main]/Main/Nexus3_repository[number-two]/online: online changed 'true' to 'false'
Notice: Finished catalog run in 0.08 seconds
$
```



Aw yeah.

puppet apply updated the online attribute



	Name ↑	Type	Format	Status
	number-two	hosted	maven2	Offline
	repo_1	hosted	maven2	Online



Adding another attribute to the Type



```
</tcinel/Desktop/nexus3/
└── lib/
    └── puppet/
        └── provider/
            └── nexus3_repository/
                rpc.rb
        └── type/
            nexus3_repository.rb
    └── puppet_x/
Gemfile
Gemfile.lock
TALKNOTES
```

The screenshot shows a terminal window with a file tree on the left and a code editor on the right. The code editor is displaying a Ruby file named `nexus3_repository.rb`. The file contains Puppet-style code defining a type for a Nexus3 repository.

```
desc 'The name of the hosted repository'
end

newproperty(:online) do
    desc 'Determines repository availability'
    newvalues(:true, :false)
    defaultto :true
    munge { |value| Nexus3::Util.munge_boolean(value) }
end

newproperty(:blobstore) do
    desc 'Which blobstore to use for artifact storage'
    defaultto 'default'

    validate do |value|
        unless /[a-zA-Z0-9\-\_\.]+/ =~ value
            fail("blobstore: Only non-empty strings containing alphanumeric
, hyphens and underscores are allowed (#{$value} given).")
        end
    end
end
```



Adding another attribute to the Type

custom input validation logic



The terminal shows the directory structure of a Puppet module:

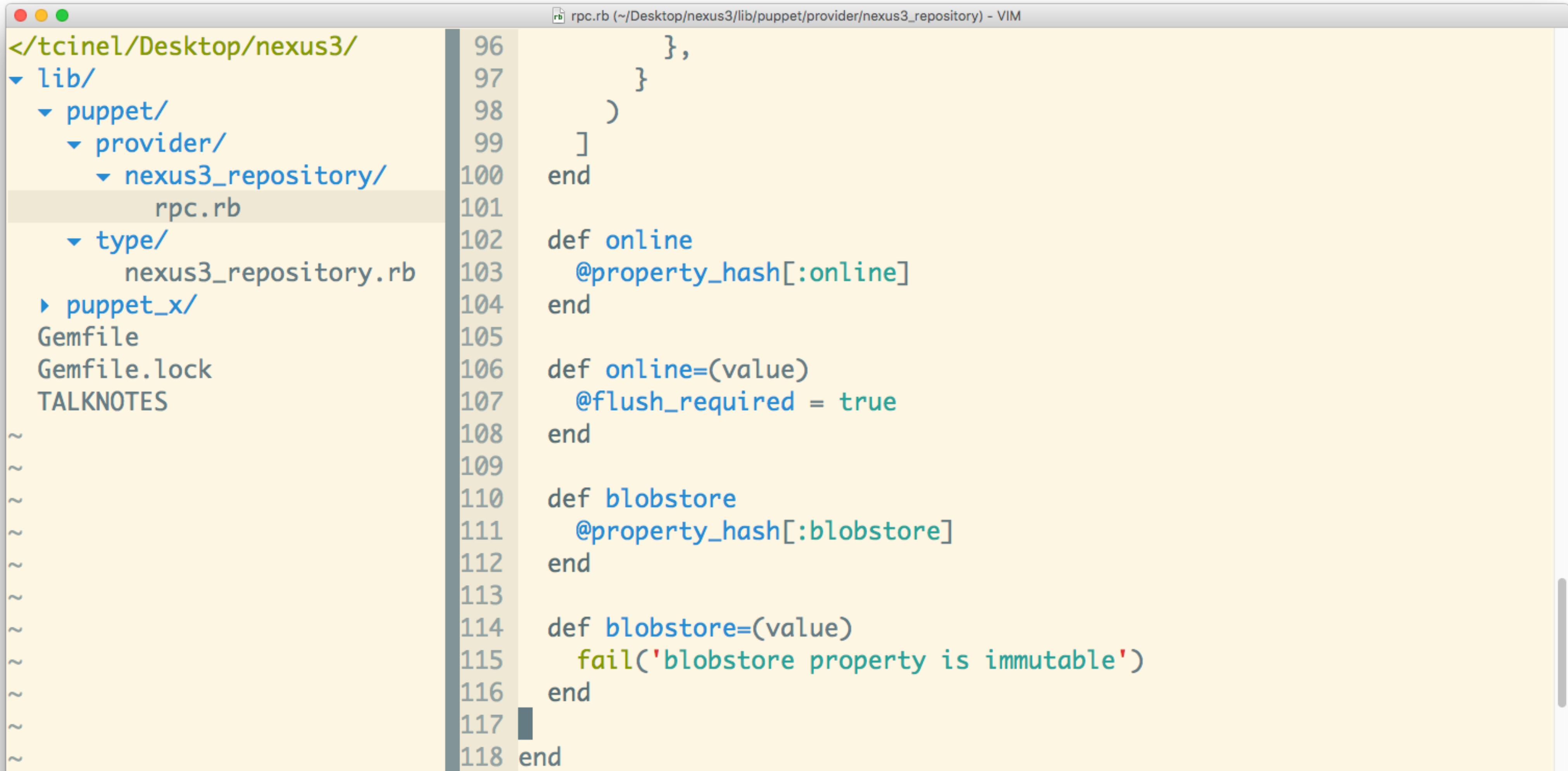
```
</tcinel/Desktop/nexus3/
└── lib/
    └── puppet/
        ├── provider/
        │   └── nexus3_repository/
        │       └── rpc.rb
        └── type/
            └── nexus3_repository.rb
```

The `nexus3_repository.rb` file in the `type/` directory is open in VIM. The code defines two properties: `:online` and `:blobstore`, and includes validation logic for the `:blobstore` attribute.

```
9  desc 'The name of the hosted repository'
10 end
11
12 newproperty(:online) do
13     desc 'Determines repository availability'
14     newvalues(:true, :false)
15     defaultto :true
16     munge { |value| Nexus3::Util.munge_boolean(value) }
17 end
18
19 newproperty(:blobstore) do
20     desc 'Which blobstore to use for artifact storage'
21     defaultto 'default'
22
23     validate do |value|
24         unless /[a-zA-Z0-9\-\_\.]+/ =~ value
25             fail("blobstore: Only non-empty strings containing alphanumeric
, hyphens and underscores are allowed (#{$value} given).")
26         end
27     end
28 end
29
```



Adding another attribute to the Provider



The image shows a file browser on the left and a Vim editor window on the right.

File Browser:

- </tcinel/Desktop/nexus3/
- lib/
 - puppet/
 - provider/
 - nexus3_repository/
 - rpc.rb
 - type/
 - nexus3_repository.rb
 - puppet_x/
 - Gemfile
 - Gemfile.lock
 - TALKNOTES

VIM Editor:

```
rb rpc.rb (~/Desktop/nexus3/lib/puppet/provider/nexus3_repository) - VIM
```

Code content:

```
96     },
97     }
98   )
99 ]
100 end
101
102 def online
103   @property_hash[:online]
104 end
105
106 def online=(value)
107   @flush_required = true
108 end
109
110 def blobstore
111   @property_hash[:blobstore]
112 end
113
114 def blobstore=(value)
115   fail('blobstore property is immutable')
116 end
117
118 end
```



Adding another attribute to the Provider

immutable property using fail method

```
</tcinel/Desktop/nexus3/
lib/
  puppet/
    provider/
      nexus3_repository/
        rpc.rb
      type/
        nexus3_repository.rb
  puppet_x/
Gemfile
Gemfile.lock
TALKNOTES

96   },
97   }
98   )
99   ]
100 end
101
102 def online
103   @property_hash[:online]
104 end
105
106 def online=(value)
107   @flush_required = true
108 end
109
110 def blobstore
111   @property_hash[:blobstore]
112 end
113
114 def blobstore=(value)
115   fail('blobstore property is immutable')
116 end
117
118 end
```



puppet resource is reporting blobstore value

```
2. bash (bash)
$ puppet resource nexus3_repository
nexus3_repository { 'number-two':
  ensure      => 'present',
  blobstore   => 'default',
  online      => 'false',
}
nexus3_repository { 'repo_1':
  ensure      => 'present',
  blobstore   => 'default',
  online      => 'true',
}
$ █
```



puppet resource is reporting blobstore value

```
$ puppet resource nexus3_repository
nexus3_repository { 'number-two':
  ensure  => 'present',
  blobstore => 'default',
  online   => 'false',
}
nexus3_repository { 'repo_1':
  ensure  => 'present',
  blobstore => 'default',
  online   => 'true',
}
$
```



changing the blobstore value for an existing repository is failing

```
2. bash (bash)
$ puppet apply -e "
> resources { 'nexus3_repository': purge => true }
> nexus3_repository {'repo_1': ensure => present, online => true, blobstore => 'default' }
> nexus3_repository {'number-two': ensure => present, online => false, blobstore => 'non-default' }
>
Notice: Compiled catalog for tcinel.home.timcinel.com in environment production in 0.07 seconds
Error: blobstore property is immutable
Error: /Stage[main]/Main/Nexus3_repository[number-two]/blobstore: change from default to non-default failed: blobstore property is immutable
Notice: Finished catalog run in 0.06 seconds
$ 
```



changing the blobstore value for an existing repository is failing

```
2. bash (bash)
$ puppet apply -e "
> resources { 'nexus3_repository': purge => true }
> nexus3_repository {'repo_1': ensure => present, online => true, blobstore => 'default' }
> nexus3_repository {'number-two': ensure => present, online => false, blobstore => 'non-default' }
>
Notice: Compiled catalog for tcinel.home.timcinel.com in environment production in 0.07 seconds
Error: blobstore property is immutable
Error: /Stage[main]/Main/Nexus3_repository[number-two]/blobstore: change from default to non-default failed: blobstore property is immutable
Notice: Finished catalog run in 0.06 seconds
$
```



That's
nexus3_repository
done.



We're over the hump.



"Silbury Hill" (CC BY 2.0) by rightee

Worked Example

SET UP

IMPLEMENT A TYPE

WRITE A PROVIDER

ADDING MORE PROPERTIES

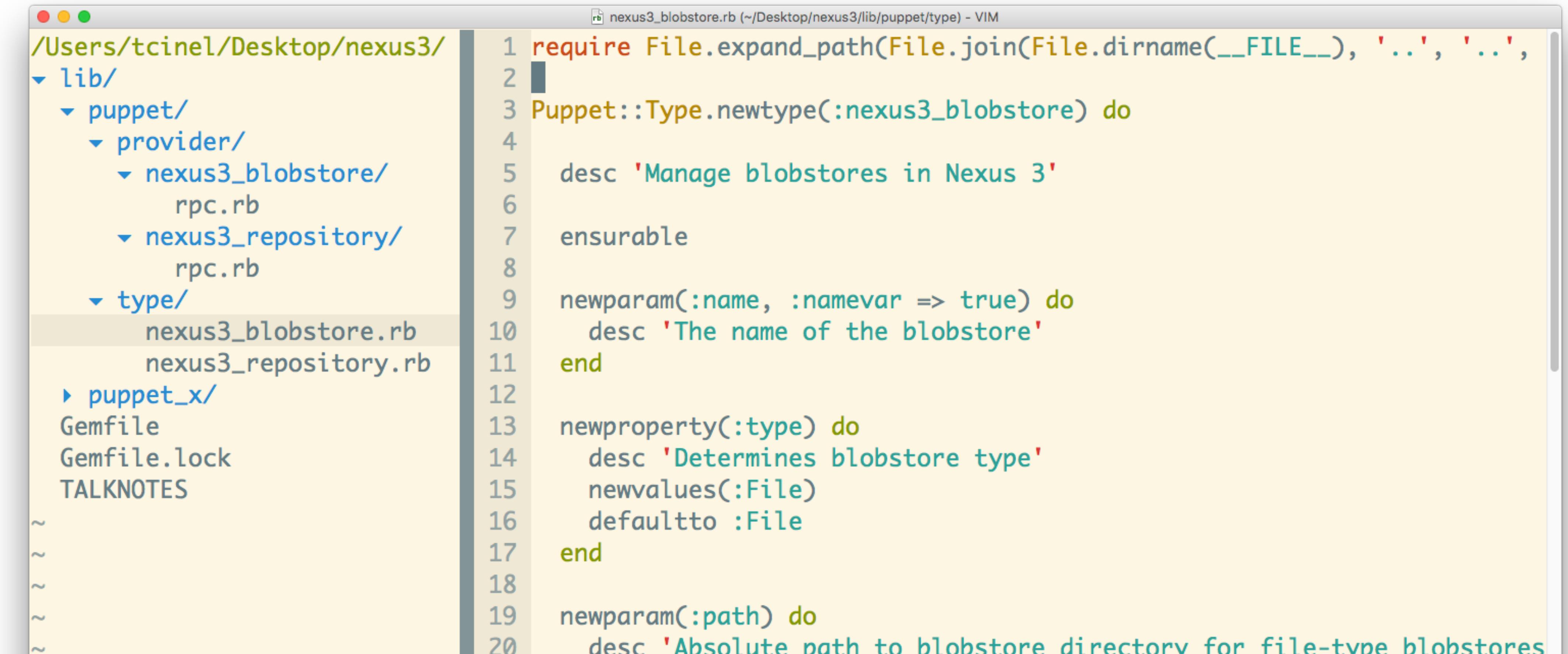
INTERACTION BETWEEN TYPES



We've created another Type

This one is called `nexus3_blobstore`

We won't walk through it like we did for `nexus3_repository`.



The screenshot shows a Mac OS X desktop environment. On the left, a terminal window displays the file structure of a 'nexus3' project located on the Desktop. The structure includes 'lib/puppet/provider/nexus3_blobstore/rpc.rb', 'lib/puppet/provider/nexus3_repository/rpc.rb', and 'lib/puppet/type/nexus3_blobstore.rb'. Other files visible include 'Gemfile', 'Gemfile.lock', and 'TALKNOTES'. On the right, a VIM editor window titled 'nexus3_blobstore.rb (~/Desktop/nexus3/lib/puppet/type) - VIM' shows the Ruby code for the 'nexus3_blobstore' type. The code defines a newtype with parameters for name, type, and path.

```
require File.expand_path(File.join(File.dirname(__FILE__), '..', '..'))  
Puppet::Type.newtype(:nexus3_blobstore) do  
  desc 'Manage blobstores in Nexus 3'  
  ensurable  
  newparam(:name, :namevar => true) do  
    desc 'The name of the blobstore'  
  end  
  newproperty(:type) do  
    desc 'Determines blobstore type'  
    newvalues(:File)  
    defaultto :File  
  end  
  newparam(:path) do  
    desc 'Absolute path to blobstore directory for file-type blobstores'  
  end  
end
```



Mostly good. But...

```
$ puppet apply -e "
> resources { 'nexus3_repository': purge => true }
> resources { 'nexus3_blobstore': purge => true }
> nexus3_repository {'s3-repo': ensure => present, online => true, blobstore => 's3' }
> nexus3_repository {'nfs-repo': ensure => present, online => false, blobstore => 'nfs' }
> nexus3_blobstore {'s3': ensure => present, path => '/nexus-data/blobs/not-really-s3' }
> nexus3_blobstore {'nfs': ensure => present, path => '/nexus-data/blobs/not-really-nfs' }
>
Notice: Compiled catalog for tcinel.home.timcinel.com in environment production in 0.08 seconds
Notice: /Stage[main]/Main/Nexus3_repository[nfs-repo]/ensure: created
Notice: /Stage[main]/Main/Nexus3_blobstore[nfs]/ensure: created
Notice: /Stage[main]/Main/Nexus3_repository[s3-repo]/ensure: created
Notice: /Stage[main]/Main/Nexus3_blobstore[s3]/ensure: created
Notice: Finished catalog run in 0.29 seconds
$ "
```



```
2. bash (bash)
$ puppet apply -e "
> resources { 'nexus3_repository': purge => true }
> resources { 'nexus3_blobstore': purge => true }
> nexus3_repository {'s3-repo': ensure => present, online => true, blobstore => 's3' }
> nexus3_repository {'nfs-repo': ensure => present, online => false, blobstore => 'nfs' }
> nexus3_blobstore {'s3': ensure => present, path => '/nexus-data/blobs/not-really-s3' }
> nexus3_blobstore {'nfs': ensure => present, path => '/nexus-data/blobs/not-really-nfs' }
> "
Notice: Compiled catalog for tcinel.home.timcinel.com in environment production in 0.08 seconds
Notice: /Stage[main]/Main/Nexus3_repository[nfs-repo]/ensure: created
Notice: /Stage[main]/Main/Nexus3_blobstore[nfs]/ensure: created
Notice: /Stage[main]/Main/Nexus3_repository[s3-repo]/ensure: created
Notice: /Stage[main]/Main/Nexus3_blobstore[s3]/ensure: created
Notice: Finished catalog run in 0.29 seconds
$ 
```



```
$ puppet apply -e "
> resources { 'nexus3_repository': purge => true }
> resources { 'nexus3_blobstore': purge => true }
> nexus3_repository {'s3-repo': ensure => present, online => true, blobstore => 's3' }
> nexus3_repository {'nfs-repo': ensure => present, online => false, blobstore => 'nfs' }
> nexus3_blobstore {'s3': ensure => present, path => '/nexus-data/blobs/not-really-s3' }
> nexus3_blobstore {'nfs': ensure => present, path => '/nexus-data/blobs/not-really-nfs' }
>
Notice: Compiled catalog for tcinel.home.timcinel.com in environment production in 0.08 seconds
Notice: /Stage[main]/Main/Nexus3_repository[nfs-repo]/ensure: created
Notice: /Stage[main]/Main/Nexus3_blobstore[nfs]/ensure: created
Notice: /Stage[main]/Main/Nexus3_repository[s3-repo]/ensure: created
Notice: /Stage[main]/Main/Nexus3_blobstore[s3]/ensure: created
Notice: Finished catalog run in 0.29 seconds
$ "
```



**nexus3_repositorys instances are being created before
nexus3_blobstore instances. not good.**

```
$ puppet apply -e "
> resources { 'nexus3_repository': purge => true }
> resources { 'nexus3_blobstore': purge => true }
> nexus3_repository {'s3-repo': ensure => present, online => true, blobstore => 's3' }
> nexus3_repository {'nfs-repo': ensure => present, online => false, blobstore => 'nfs' }
> nexus3_blobstore {'s3': ensure => present, path => '/nexus-data/blobs/not-really-s3' }
> nexus3_blobstore {'nfs': ensure => present, path => '/nexus-data/blobs/not-really-nfs' }
>
Notice: Compiled catalog for tcinel.home.timcinel.com in environment production in 0.08 seconds
Notice: /Stage[main]/Main/Nexus3_repository[nfs-repo]/ensure: created
Notice: /Stage[main]/Main/Nexus3_blobstore[nfs]/ensure: created
Notice: /Stage[main]/Main/Nexus3_repository[s3-repo]/ensure: created
Notice: /Stage[main]/Main/Nexus3_blobstore[s3]/ensure: created
Notice: Finished catalog run in 0.29 seconds
$ "
```



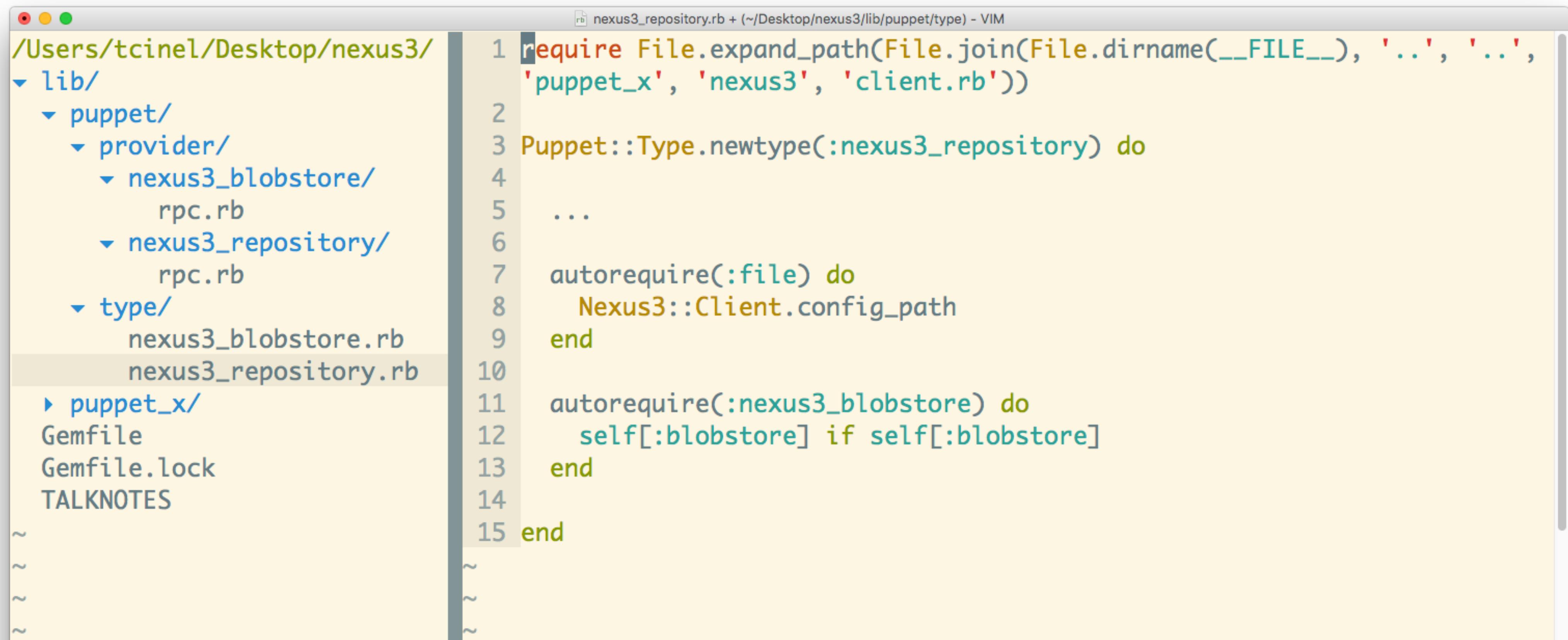
**nexus3_repository instances are being created before
nexus3_blobstore instances. not good.**

```
$ puppet apply -e "
> resources { 'nexus3_repository': purge => true }
> resources { 'nexus3_blobstore': purge => true }
> nexus3_repository {'s3-repo': ensure => present, online => true, blobstore => 's3' }
> nexus3_repository {'nfs-repo': ensure => present, online => false, blobstore => 'nfs' }
> nexus3_blobstore {'s3': ensure => present, path => '/nexus-data/blobs/not-really-s3' }
> nexus3_blobstore {'nfs': ensure => present, path => '/nexus-data/blobs/not-really-nfs' }
>
Notice: Compiled catalog for tcinel.home.timcinel.com in environment production in 0.08 seconds
Notice: /Stage[main]/Main/Nexus3_repository[nfs-repo]/ensure: created
Notice: /Stage[main]/Main/Nexus3_blobstore[nfs]/ensure: created
Notice: /Stage[main]/Main/Nexus3_repository[s3-repo]/ensure: created
Notice: /Stage[main]/Main/Nexus3_blobstore[s3]/ensure: created
Notice: Finished catalog run in 0.29 seconds
$ "
```



Relationships between Types

Types can define *soft require* relationships with arbitrary resources



The screenshot shows a terminal window with a file tree on the left and a Vim editor on the right.

File Tree:

```
/Users/tcinel/Desktop/nexus3/
└── lib/
    ├── puppet/
    │   ├── provider/
    │   │   └── nexus3_blobstore/
    │   │       └── rpc.rb
    │   └── type/
    │       ├── nexus3_blobstore.rb
    │       └── nexus3_repository.rb
    └── puppet_x/
        └── Gemfile
        └── Gemfile.lock
    TALKNOTES
```

Vim Editor: The file being edited is `nexus3_repository.rb`.

```
require File.expand_path(File.join(File.dirname(__FILE__), '..', '..', 'puppet_x', 'nexus3', 'client.rb'))

Puppet::Type.newtype(:nexus3_repository) do
  ...
  autorequire(:file) do
    Nexus3::Client.config_path
  end
  autorequire(:nexus3_blobstore) do
    self[:blobstore] if self[:blobstore]
  end
end
```



Relationships between Types

Types can define *soft require* relationships with arbitrary resources

if puppet manages this config file, then all nexus3_repository resources will depend on it



The image shows a terminal window with two panes. The left pane displays a file tree for a directory named 'nexus3'. The right pane shows a Ruby code editor with a file named 'nexus3_repository.rb'. The code defines a new type 'nexus3_repository' and includes an 'autorequire' statement for a file resource.

```
require File.expand_path(File.join(File.dirname(__FILE__), '..', '..', 'puppet_x', 'nexus3', 'client.rb'))  
Puppet::Type.newtype(:nexus3_repository) do  
  ...  
  autorequire(:file) do  
    Nexus3::Client.config_path  
  end  
  autorequire(:nexus3_blobstore) do  
    self[:blobstore] if self[:blobstore]  
  end  
end
```



Relationships between Types

Types can define *soft require* relationships with arbitrary resources

the associated `nexus3_blobstore` resource will become a dependency (if it exists)



The image shows a terminal window with a file tree on the left and a Vim editor window on the right.

File Tree:

- /Users/tcinel/Desktop/nexus3/
- lib/
- puppet/
- provider/
- nexus3_blobstore/
- rpc.rb
- nexus3_repository/
- rpc.rb
- type/
- nexus3_blobstore.rb
- nexus3_repository.rb

VIM Editor (nexus3_repository.rb):

```
1 require File.expand_path(File.join(File.dirname(__FILE__), '..', '..', 'puppet_x', 'nexus3', 'client.rb'))
2
3 Puppet::Type.newtype(:nexus3_repository) do
4
5   ...
6
7   autorequire(:file) do
8     Nexus3::Client.config_path
9   end
10
11  autorequire(:nexus3_blobstore) do
12    self[:blobstore] if self[:blobstore]
13  end
14
15 end
```

A yellow rectangular highlight covers the code from line 11 to line 13, specifically the block `autorequire(:nexus3_blobstore) do` and its condition `self[:blobstore]`.



Better.

```
2. bash (bash)
$ puppet apply -e "
> resources { 'nexus3_repository': purge => true }
> resources { 'nexus3_blobstore': purge => true }
> nexus3_repository {'s3-repo': ensure => present, online => true, blobstore => 's3' }
> nexus3_repository {'nfs-repo': ensure => present, online => false, blobstore => 'nfs' }
> nexus3_blobstore {'s3': ensure => present, path => '/nexus-data/blobs/not-really-s3' }
> nexus3_blobstore {'nfs': ensure => present, path => '/nexus-data/blobs/not-really-nfs' }
>

Notice: Compiled catalog for tcinel.home.timcinel.com in environment production in 0.09 seconds
Notice: /Stage[main]/Main/Nexus3_blobstore[nfs]/ensure: created
Notice: /Stage[main]/Main/Nexus3_repository[nfs-repo]/ensure: created
Notice: /Stage[main]/Main/Nexus3_blobstore[s3]/ensure: created
Notice: /Stage[main]/Main/Nexus3_repository[s3-repo]/ensure: created
Notice: Finished catalog run in 0.28 seconds
$
```



blobstore created before each dependent repository

```
2. bash (bash)
$ puppet apply -e "
> resources { 'nexus3_repository': purge => true }
> resources { 'nexus3_blobstore': purge => true }
> nexus3_repository {'s3-repo': ensure => present, online => true, blobstore => 's3' }
> nexus3_repository {'nfs-repo': ensure => present, online => false, blobstore => 'nfs' }
> nexus3_blobstore {'s3': ensure => present, path => '/nexus-data/blobs/not-really-s3' }
> nexus3_blobstore {'nfs': ensure => present, path => '/nexus-data/blobs/not-really-nfs' }
> "

Notice: Compiled catalog for tcinel.home.timcinel.com in environment production in 0.09 seconds
Notice: /Stage[main]/Main/Nexus3_blobstore[nfs]/ensure: created
Notice: /Stage[main]/Main/Nexus3_repository[nfs-repo]/ensure: created
Notice: /Stage[main]/Main/Nexus3_blobstore[s3]/ensure: created
Notice: /Stage[main]/Main/Nexus3_repository[s3-repo]/ensure: created
Notice: Finished catalog run in 0.28 seconds
$
```



Even better.

```
2. bash (bash)
$ rm /Users/tcinel/.puppet/nexus3_conf.yaml
$ puppet apply -e "
resources { 'nexus3_repository': purge => true }
resources { 'nexus3_blobstore': purge => true }
file { '/Users/tcinel/.puppet/nexus3_conf.yaml': source => '/Users/tcinel/tmp/nexus3_conf.yaml' }
nexus3_repository {'s3-repo': ensure => present, online => true, blobstore => 's3' }
nexus3_repository {'nfs-repo': ensure => present, online => false, blobstore => 'nfs' }
nexus3_blobstore {'s3': ensure => present, path => '/nexus-data/blobs/not-really-s3' }
nexus3_blobstore {'nfs': ensure => present, path => '/nexus-data/blobs/not-really-nfs' }
"
Notice: Compiled catalog for tcinel.home.timcinel.com in environment production in 0.20 seconds
Error: /Stage[main]/Main/Resources[nexus3_repository]: Failed to generate additional resources using 'generate': No such file or directory - /Users/tcinel/.puppet/nexus3_conf.yaml
Error: /Stage[main]/Main/Resources[nexus3_blobstore]: Failed to generate additional resources using 'generate': No such file or directory - /Users/tcinel/.puppet/nexus3_conf.yaml
Notice: /Stage[main]/Main/File[/Users/tcinel/.puppet/nexus3_conf.yaml]/ensure: defined content as '{md5}01
fef7b0507b61dd6af5b2a4718933e9'
Notice: /Stage[main]/Main/Nexus3_blobstore[nfs]/ensure: created
Notice: /Stage[main]/Main/Nexus3_repository[nfs-repo]/ensure: created
Notice: /Stage[main]/Main/Nexus3_blobstore[s3]/ensure: created
Notice: /Stage[main]/Main/Nexus3_repository[s3-repo]/ensure: created
Notice: Finished catalog run in 0.34 seconds
$ 
```



allow Puppet to manage the config file

```
2. bash (bash)
$ rm /Users/tcinel/.puppet/nexus3_conf.yaml
$ puppet apply -e "
resources { 'nexus3_repository': purge => true }
resources { 'nexus3_blobstore': purge => true }
file { '/Users/tcinel/.puppet/nexus3_conf.yaml': source => '/Users/tcinel/tmp/nexus3_conf.yaml' }
nexus3_repository {'s3-repo': ensure => present, online => true, blobstore => 's3' }
nexus3_repository {'nfs-repo': ensure => present, online => false, blobstore => 'nfs' }
nexus3_blobstore {'s3': ensure => present, path => '/nexus-data/blobs/not-really-s3' }
nexus3_blobstore {'nfs': ensure => present, path => '/nexus-data/blobs/not-really-nfs' }
"
Notice: Compiled catalog for tcinel.home.timcinel.com in environment production in 0.20 seconds
Error: /Stage[main]/Main/Resources[nexus3_repository]: Failed to generate additional resources using 'generate': No such file or directory - /Users/tcinel/.puppet/nexus3_conf.yaml
Error: /Stage[main]/Main/Resources[nexus3_blobstore]: Failed to generate additional resources using 'generate': No such file or directory - /Users/tcinel/.puppet/nexus3_conf.yaml
Notice: /Stage[main]/Main/File[/Users/tcinel/.puppet/nexus3_conf.yaml]/ensure: defined content as '{md5}01
fef7b0507b61dd6af5b2a4718933e9'
Notice: /Stage[main]/Main/Nexus3_blobstore[nfs]/ensure: created
Notice: /Stage[main]/Main/Nexus3_repository[nfs-repo]/ensure: created
Notice: /Stage[main]/Main/Nexus3_blobstore[s3]/ensure: created
Notice: /Stage[main]/Main/Nexus3_repository[s3-repo]/ensure: created
Notice: Finished catalog run in 0.34 seconds
$
```



prefetch fails due to missing provider configuration file

```
2. bash (bash)
$ rm /Users/tcinel/.puppet/nexus3_conf.yaml
$ puppet apply -e "
resources { 'nexus3_repository': purge => true }
resources { 'nexus3_blobstore': purge => true }
file { '/Users/tcinel/.puppet/nexus3_conf.yaml': source => '/Users/tcinel/tmp/nexus3_conf.yaml' }
nexus3_repository {'s3-repo': ensure => present, online => true, blobstore => 's3' }
nexus3_repository {'nfs-repo': ensure => present, online => false, blobstore => 'nfs' }
nexus3_blobstore {'s3': ensure => present, path => '/nexus-data/blobs/not-really-s3' }
nexus3_blobstore {'nfs': ensure => present, path => '/nexus-data/blobs/not-really-nfs' }
"
Notice: Compiled catalog for tcinel.home.timcinel.com in environment production in 0.20 seconds
Error: /Stage[main]/Main/Resources[nexus3_repository]: Failed to generate additional resources using 'generate': No such file or directory - /Users/tcinel/.puppet/nexus3_conf.yaml
Error: /Stage[main]/Main/Resources[nexus3_blobstore]: Failed to generate additional resources using 'generate': No such file or directory - /Users/tcinel/.puppet/nexus3_conf.yaml
Notice: /Stage[main]/Main/File[/Users/tcinel/.puppet/nexus3_conf.yaml]/ensure: defined content as '{md5}01
fef7b0507b61dd6af5b2a4718933e9'
Notice: /Stage[main]/Main/Nexus3_blobstore[nfs]/ensure: created
Notice: /Stage[main]/Main/Nexus3_repository[nfs-repo]/ensure: created
Notice: /Stage[main]/Main/Nexus3_blobstore[s3]/ensure: created
Notice: /Stage[main]/Main/Nexus3_repository[s3-repo]/ensure: created
Notice: Finished catalog run in 0.34 seconds
$
```



Puppet creates config file, then succeeds to create resources in the right order using our custom types and providers

```
2. bash (bash)
$ rm /Users/tcinel/.puppet/nexus3_conf.yaml
$ puppet apply -e "
resources { 'nexus3_repository': purge => true }
resources { 'nexus3_blobstore': purge => true }
file { '/Users/tcinel/.puppet/nexus3_conf.yaml': source => '/Users/tcinel/tmp/nexus3_conf.yaml' }
nexus3_repository {'s3-repo': ensure => present, online => true, blobstore => 's3' }
nexus3_repository {'nfs-repo': ensure => present, online => false, blobstore => 'nfs' }
nexus3_blobstore {'s3': ensure => present, path => '/nexus-data/blobs/not-really-s3' }
nexus3_blobstore {'nfs': ensure => present, path => '/nexus-data/blobs/not-really-nfs' }
"
Notice: Compiled catalog for tcinel.home.timcinel.com in environment production in 0.20 seconds
Error: /Stage/main/Main/Resources[nexus3_repository]: Failed to generate additional resources using 'generate': No such file or directory - /Users/tcinel/.puppet/nexus3_conf.yaml
Error: /Stage/main/Main/Resources[nexus3_blobstore]: Failed to generate additional resources using 'generate': No such file or directory - /Users/tcinel/.puppet/nexus3_conf.yaml
Notice: /Stage/main/Main/File[/Users/tcinel/.puppet/nexus3_conf.yaml]/ensure: defined content as '{md5}01
fef7b0507b61dd6af5b2a4718933e9'
Notice: /Stage/main/Main/Nexus3_blobstore[nfs]/ensure: created
Notice: /Stage/main/Main/Nexus3_repository[nfs-repo]/ensure: created
Notice: /Stage/main/Main/Nexus3_blobstore[s3]/ensure: created
Notice: /Stage/main/Main/Nexus3_repository[s3-repo]/ensure: created
Notice: Finished catalog run in 0.34 seconds
$
```



Worked Example

SET UP

IMPLEMENT A TYPE

WRITE A PROVIDER

ADDING MORE PROPERTIES

INTERACTION BETWEEN TYPES

Done 



Demo



Next Steps



Areas for Improvement

Robustness

Incorporate better error handling, logging.



Tests

Unit tests with RSpec, acceptance tests with Beaker and some CI.

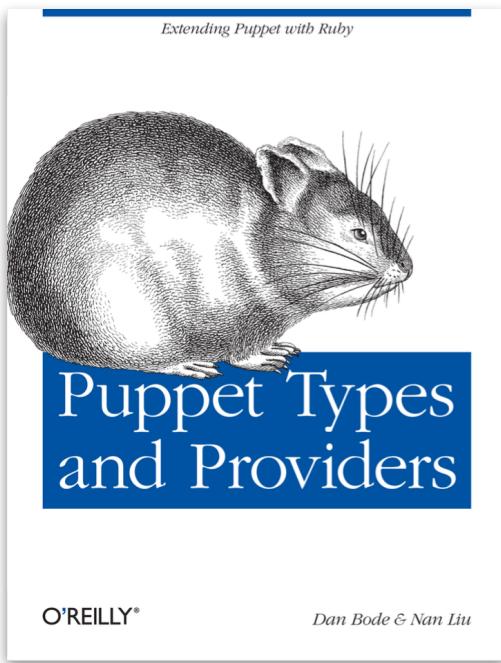


Publish to Forge

Make the module and its glorious types available to the community.



Other Resources



**Puppet Types
and Providers**



Shit Garry Says
<http://garylarizza.com/blog/>



Puppet Source



Now go, and write
some custom types
already!





Thank you

<https://bitbucket.org/TCinel/puppet-camp-example/>



TIM CINEL • BUILD WHISPERER • ATLASSIAN • @TIMCINEL