Both ProvisioningTemplate and Ptable contain overrides of the Rails base_class method, indicating that the STI base class of ProvisioningTemplate (etc) is ProvisioningTemplate, rather than Template.

```ruby
def self.base_class
  self
end
```

This causes an issue with the Template model instantiation under Rails 5.0, as ActiveRecord performs casting of new models, much like Foreman::STI does. This relies on base_class returning the actual base class, else the inheritance appears inverted and causes this exception:

```
DomainTest#test_0008_should update total_hosts on changing primary interface domain:
ActiveRecord::SubclassNotFound: Invalid single-table inheritance type: Template is not a subclass of Ptable
activerecord (5.0.0) lib/active_record/inheritance.rb:189:in `find_sti_class'
activerecord (5.0.0) lib/active_record/inheritance.rb:209:in `subclass_from_attributes'
activerecord (5.0.0) lib/active_record/inheritance.rb:58:in `new'
```

Removing base_class causes the problem noted in the comment above, that polymorphic associations to these models break - in fact, they search for the base class rather than the specific class. [https://github.com/rails/rails/blob/v5.0.0/activerecord/lib/active_record/associations.rb#L780](https://github.com/rails/rails/blob/v5.0.0/activerecord/lib/active_record/associations.rb#L780) notes that in actual fact, the base class should be stored in the type column for polymorphic associations, not the derivative.

This should be changed to remove the overrides and store the base class when dealing with polymorphic STI associations (both Taxonomy and Audit are associated to Template derivatives).

**Related issues:**
- Related to Foreman Remote Execution - Refactor #16195: Remove base_class over...
- Associated revisions
  - Revision 5908bf41 - 08/22/2016 03:39 AM - Dominic Cleal
    - fixes #16172 - remove base_class override in template subclasses

On Rails 5.0, the base_class is used during model instantiation to...
validate STI relationships as it now casts similarly to Foreman::STI, causing failures when base_class returns the subclass rather than parent class.

Removing the overriding of `base_class` in Template subclasses necessitates changing polymorphic associations to store the base class in the table's type field rather than the subclass name because the has_many associations will search by `base_class`. This is recommended by ActiveRecord when using STI with polymorphic associations:

https://github.com/rails/rails/blob/v5.0.0/activerecord/lib/active_record/associations.rb#L780

---

**History**

#1 - 08/18/2016 10:59 AM - The Foreman Bot
- Status changed from Assigned to Ready For Testing
- Pull request https://github.com/theforeman/foreman/pull/3745 added

#2 - 08/19/2016 04:04 AM - Marek Hulán
- Related to Refactor #16195: Remove base_class override in job_template added

#3 - 08/22/2016 03:40 AM - Marek Hulán
- Legacy Backlogs Release (now unused) set to 160

#4 - 08/22/2016 04:02 AM - Dominic Cleal
- Status changed from Ready For Testing to Closed
- % Done changed from 0 to 100

Applied in changeset 5908bf418d28c57dcdcd4d2e01ccf3253c0e8a81a.

03/27/2022