

# Solving Problems Fast

- You are planning a resource-demanding experiment
- You want to play around, trying different scenarios for your solution, by pressing a couple of buttons
- You want to get the results fast

# Harness the resources

- Typical High-Performance node has 8-16 cores and 8-16-32 GB of RAM
- Those nodes are powerful and easy to allocate
- But they are ... empty

# Put Node to Life

- Provision a node – install all the packages needed to solve the problem
- Loosing time and money while doing it manually?  
Potentially repeat many times?
- Use special ‘recipes’, from Puppet:
  - ✓ Install packages you need automatically and remotely
  - ✓ Reuse the same code to provision different systems

# The Case: “Bank Client Satisfaction”

<https://www.kaggle.com/c/santander-customer-satisfaction/data>

## Provisioning solution to solve the case:

- 1) The ready to use environment is being created in 5-20 minutes in a fully automated manner (instead of several hours) and can be easily recreated in the case of a node failure
- 2) The created environment provides intuitive and user-friendly Web-based interface
- 3) The solution is hosted by GitHub and can be easily installed on a *local* machine to steer the *remote* Cloud calculations
- 4) Having more time, the solution would benefit from becoming more versatile to support different OS, like MacOS, Windows, different flavours of Linux