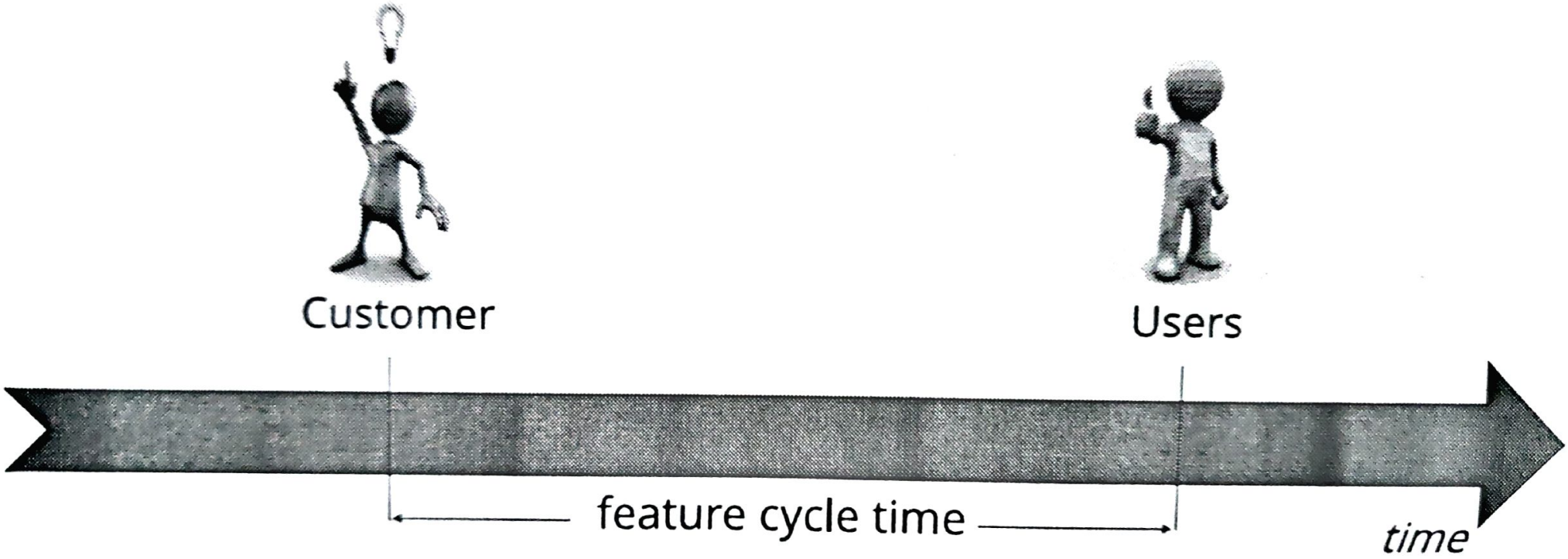
Configuration Management Tool

Ansible

By
Venkat
Rise 'n' Shine Technologies

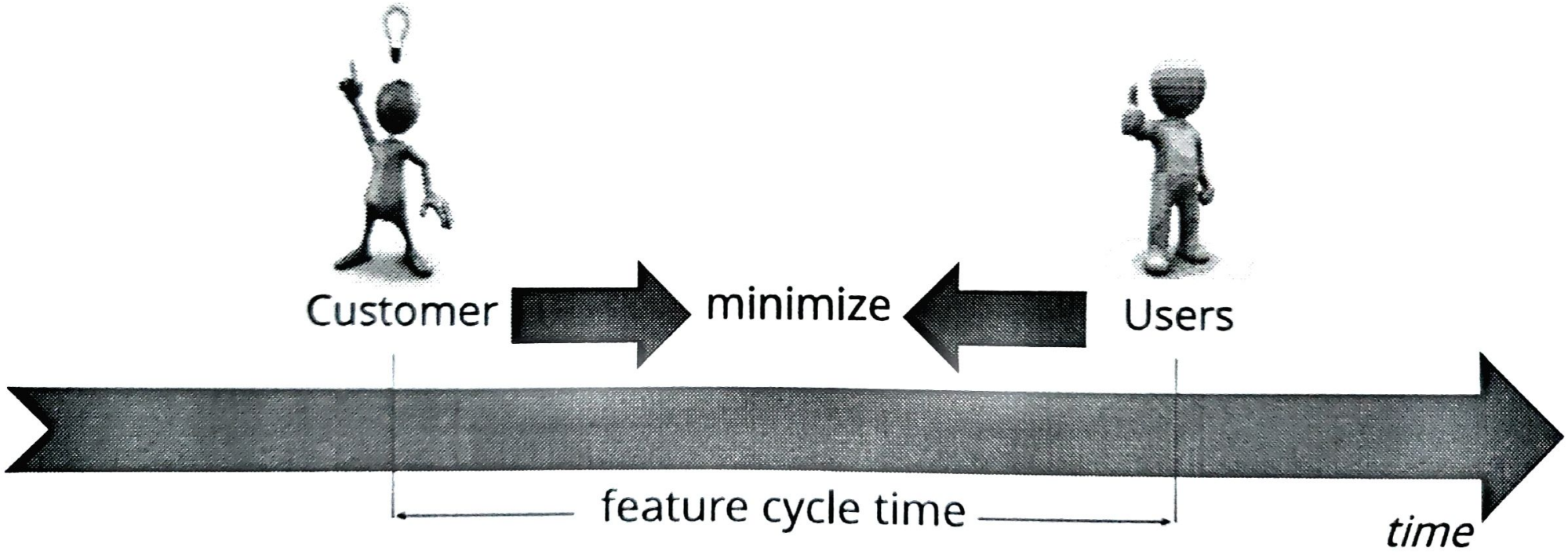
Utmost Goal: Minimize Cycle Time



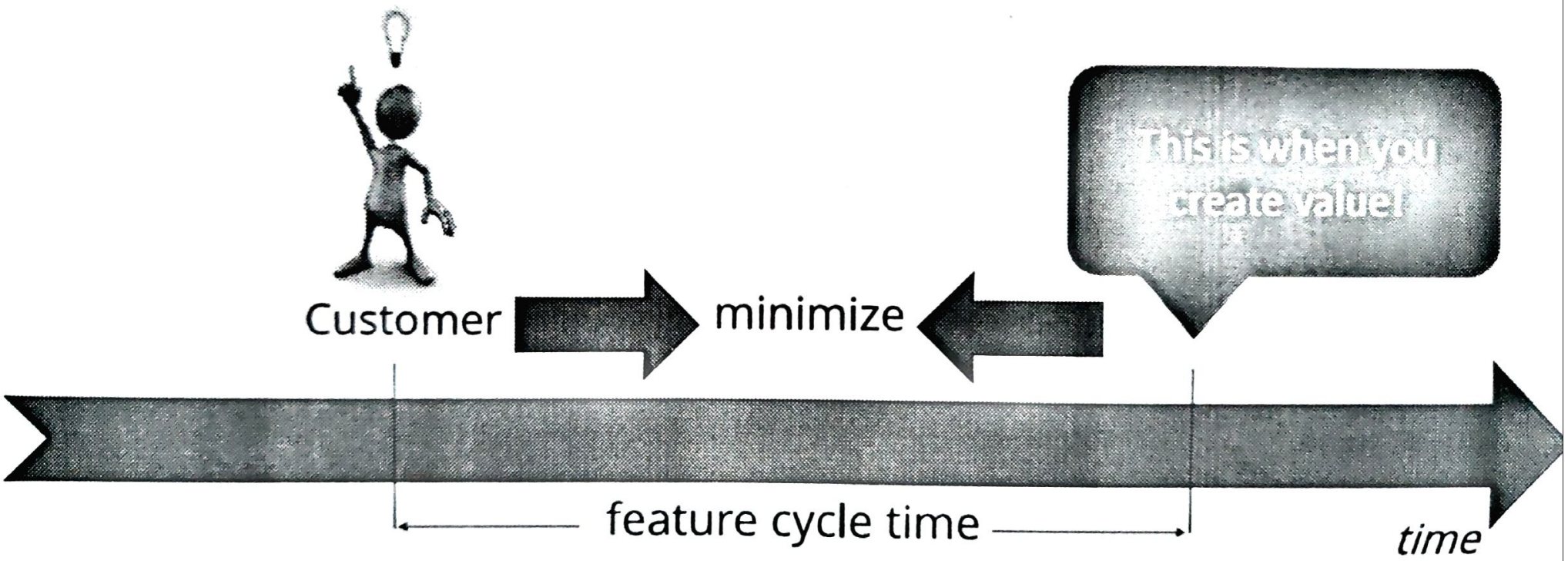
Cycle time is the most relevant metric in the software delivery process.

"How long would it take your organization to deploy a change that involves just one single line of code?

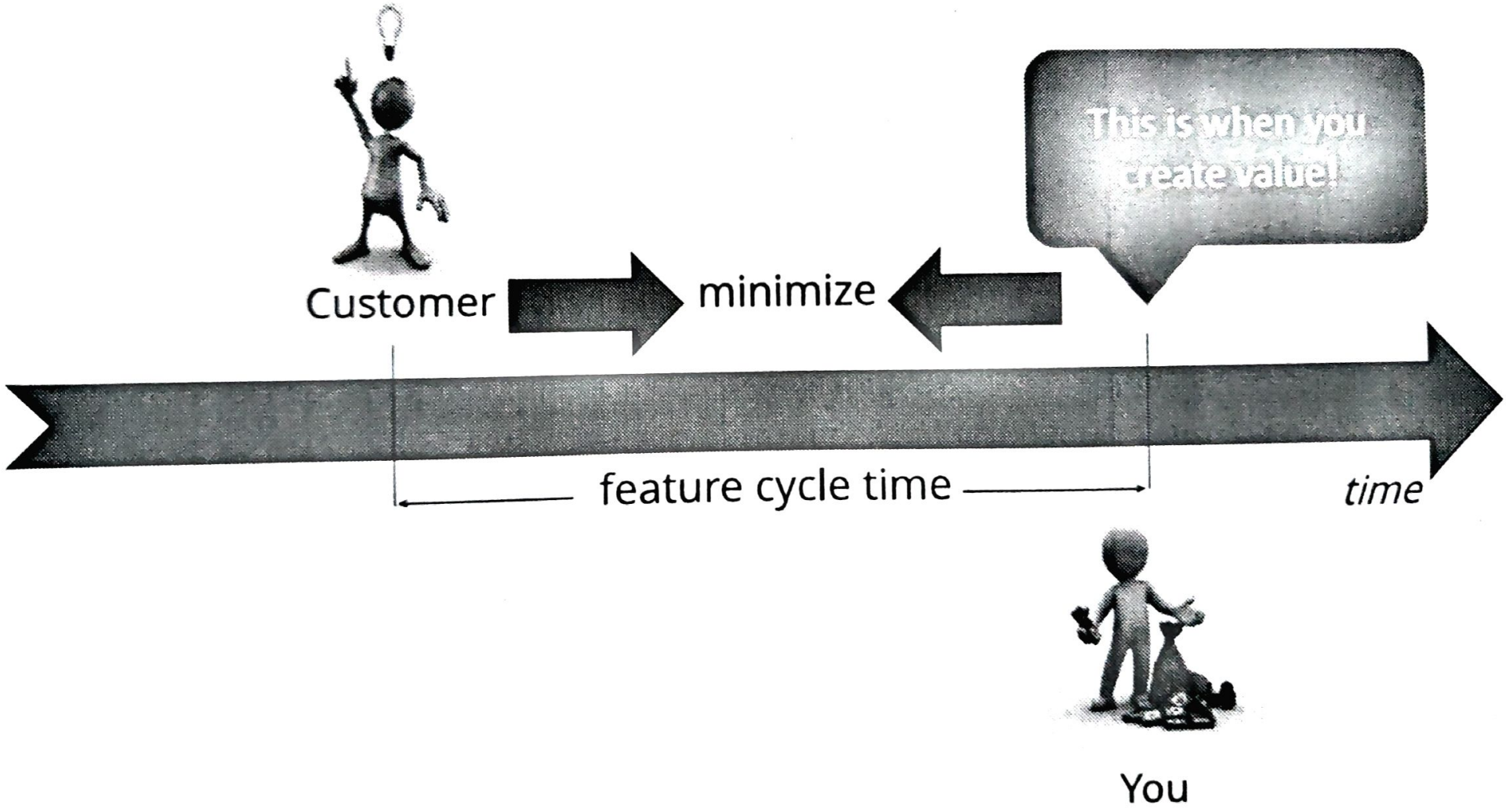
Utmost Goal: Minimize Cycle Time



Utmost Goal: Minimize Cycle Time



Utmost Goal: Minimize Cycle Time

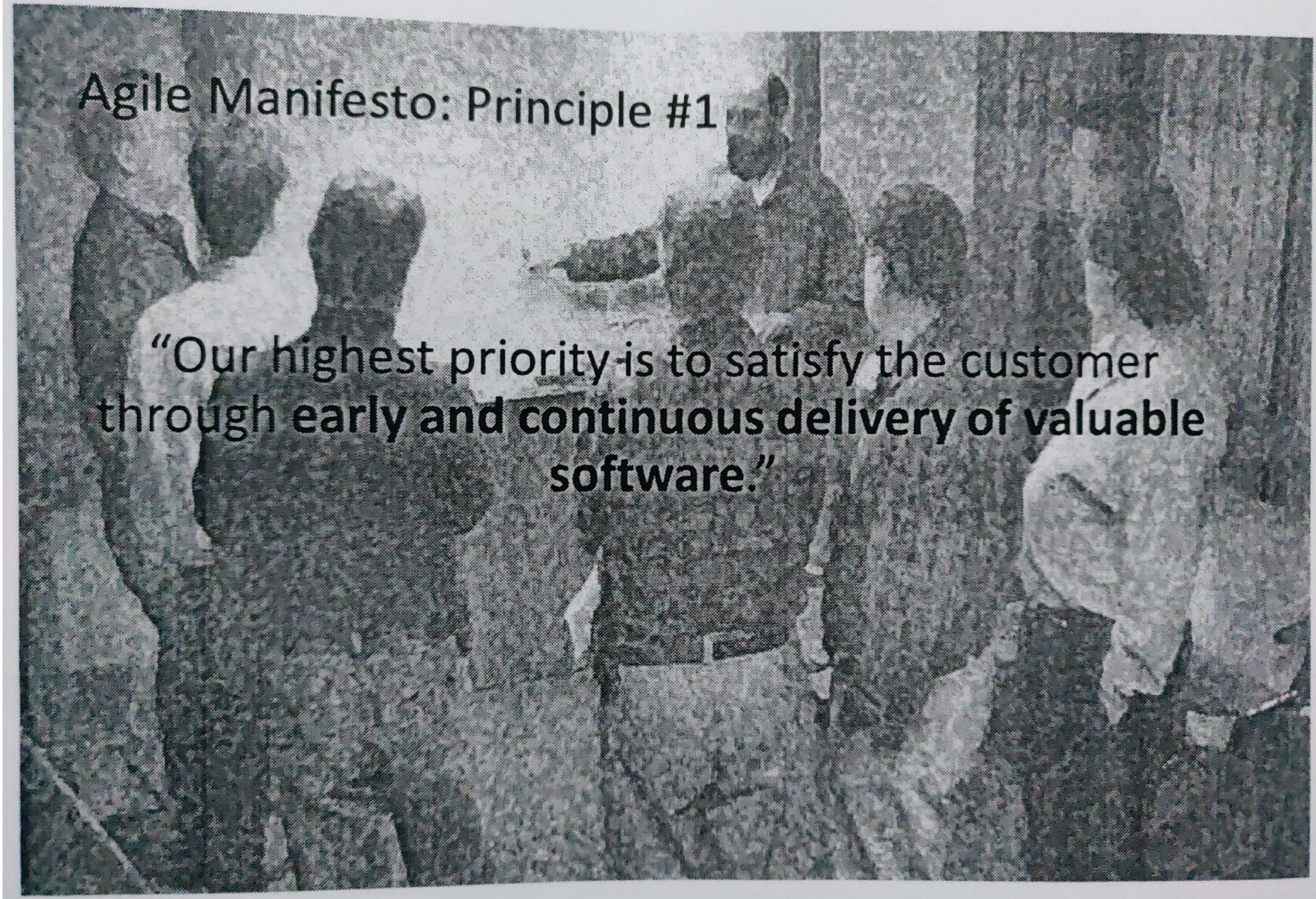


Utmost Goal: Minimize Cycle Time

It's about getting your features into your user's hands

as quickly as possible!

You

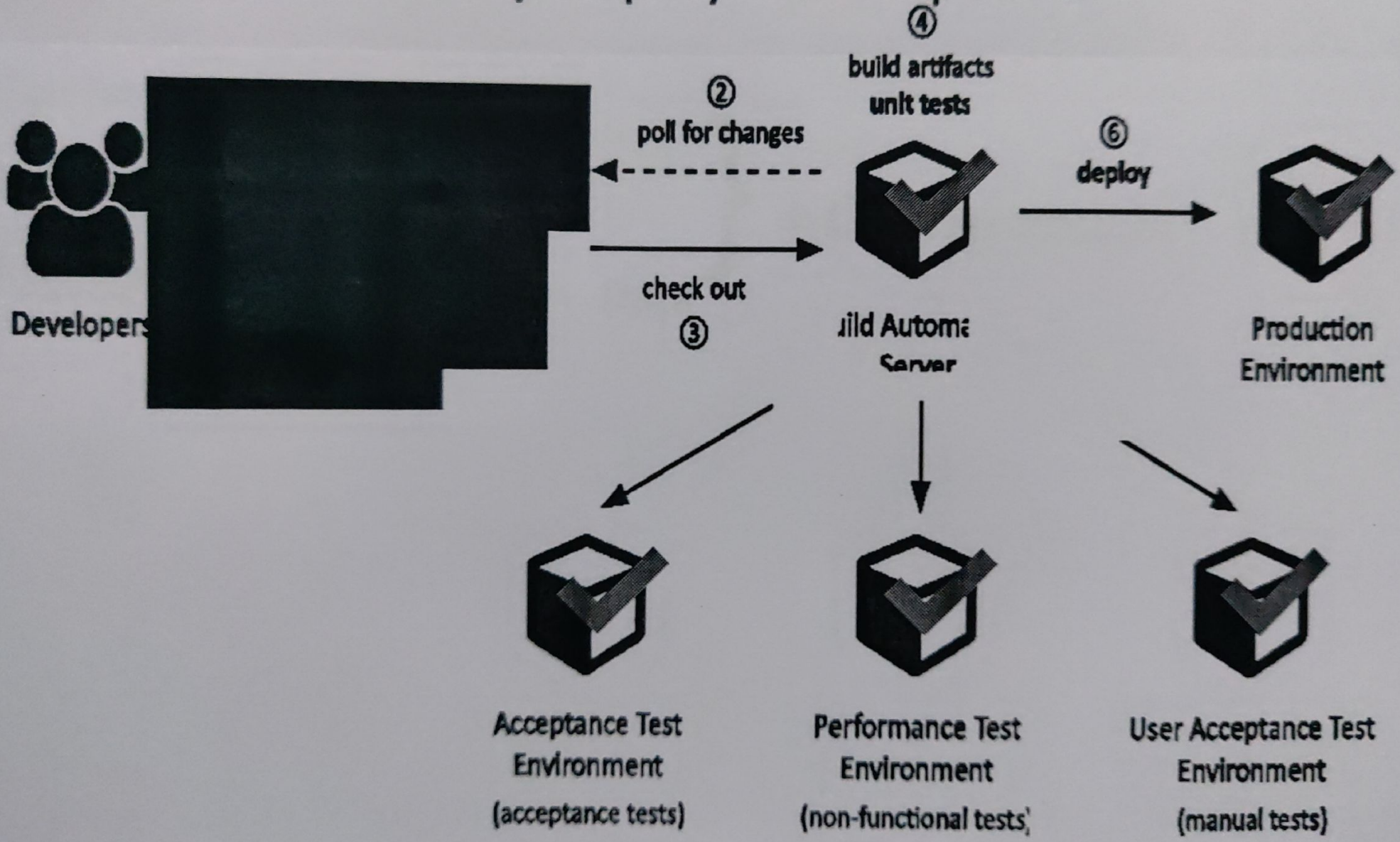

Agile Manifesto: Principle #1
"Our highest priority is to satisfy the customer through **early and continuous delivery of valuable software.**"

Continuous Delivery Deployment Pipeline

...which is at the heart of Continuous Delivery, defined as:

*"A deployment pipeline is, in essence, an **automated implementation** of your application's **build, deploy, test and release** process."*

Continuous Delivery Deployment Pipeline



Automated Deployments

Why?

» Create application runtime environments **on demand**

» **Fast**, **reliable**, **repeatable** and **predictable** outcomes

» **Consistent** environments in staging and production

» Establish **fast feedback loops** you can react upon

» Makes release days **riskless**, almost boring

Automated Deployments

What?

» Operating Systems, Drivers

» Middleware, Databases, etc.

» Applications, Dependencies, Data

} + Configuration

Automated Deployments

How?

» Infrastructure as Code!

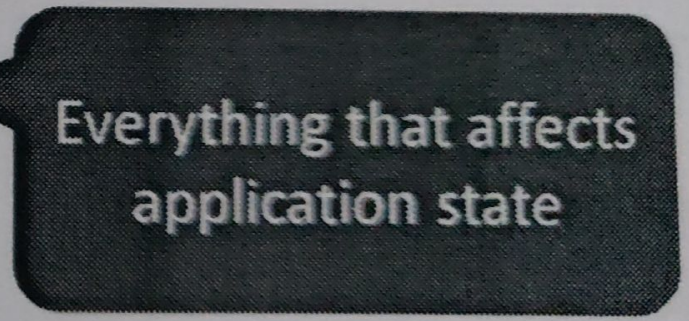
» Keep everything in Version Control

  » Code

  » Configuration

  » Data

» Align Development and Operations



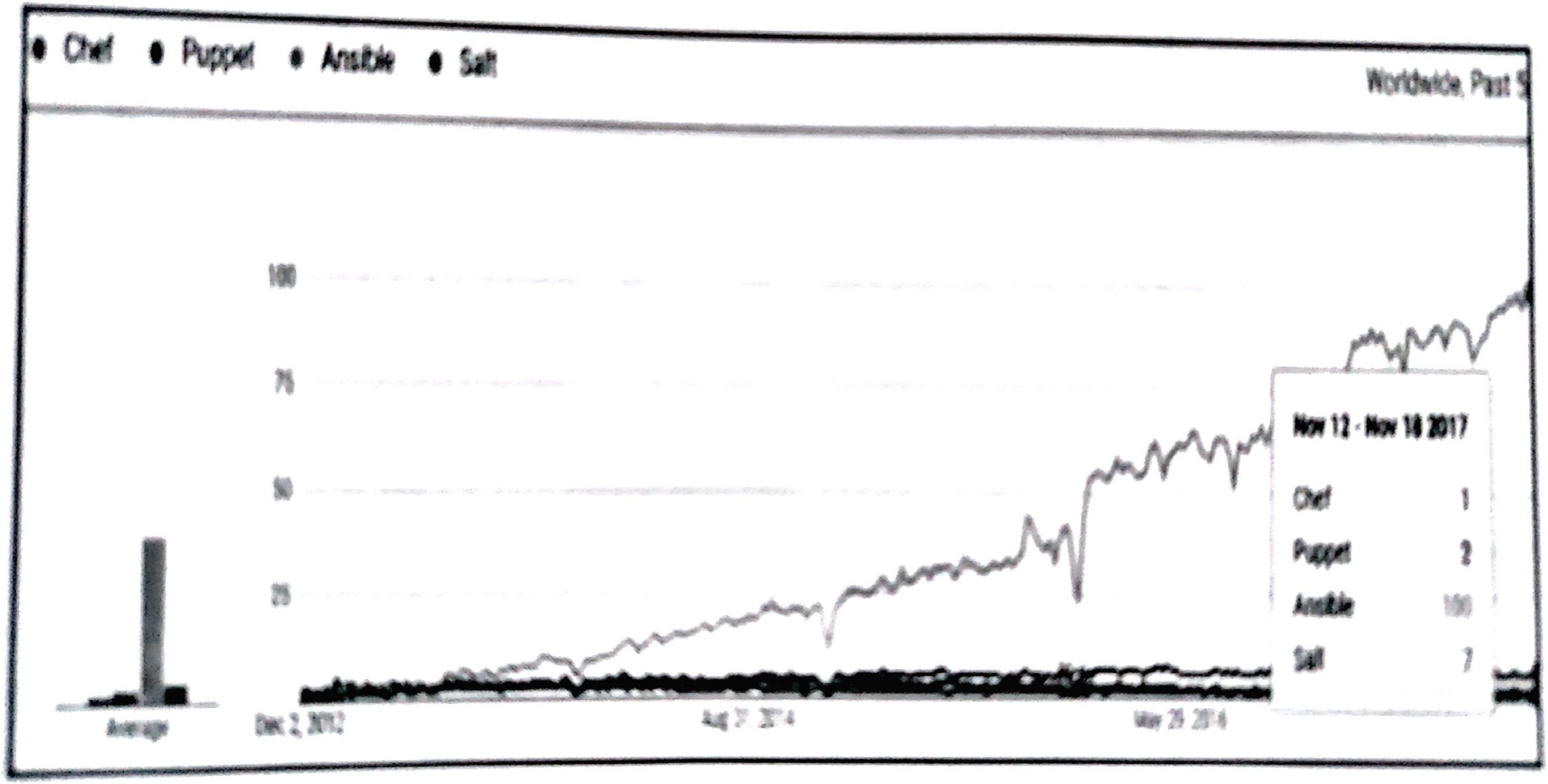
What is an Ansible?

- Ansible is, in short, an IT automation, configuration management and provisioning tool.

- It uses 'playbooks' to deploy, manage, build, test and configure anything from full server environments to websites to custom compiled source code for applications.

- It brings together aspects of environment management that have been traditionally separate and managed independently.

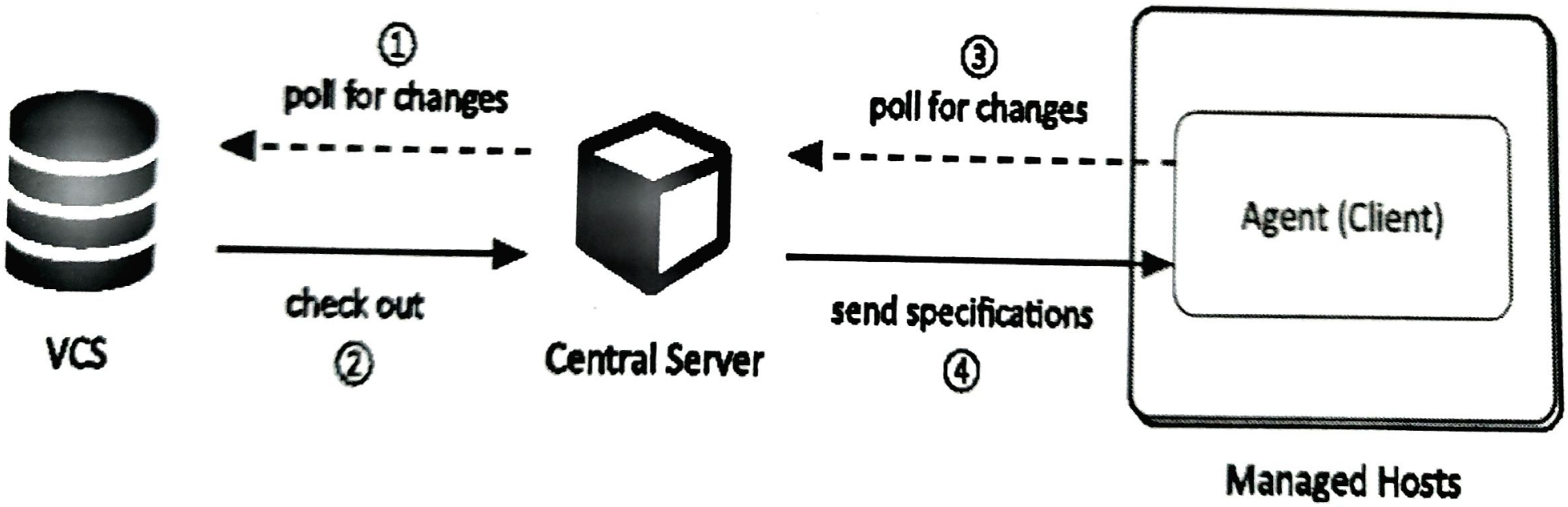
Global Google Trends: Chef vs. Puppet vs. Ansible



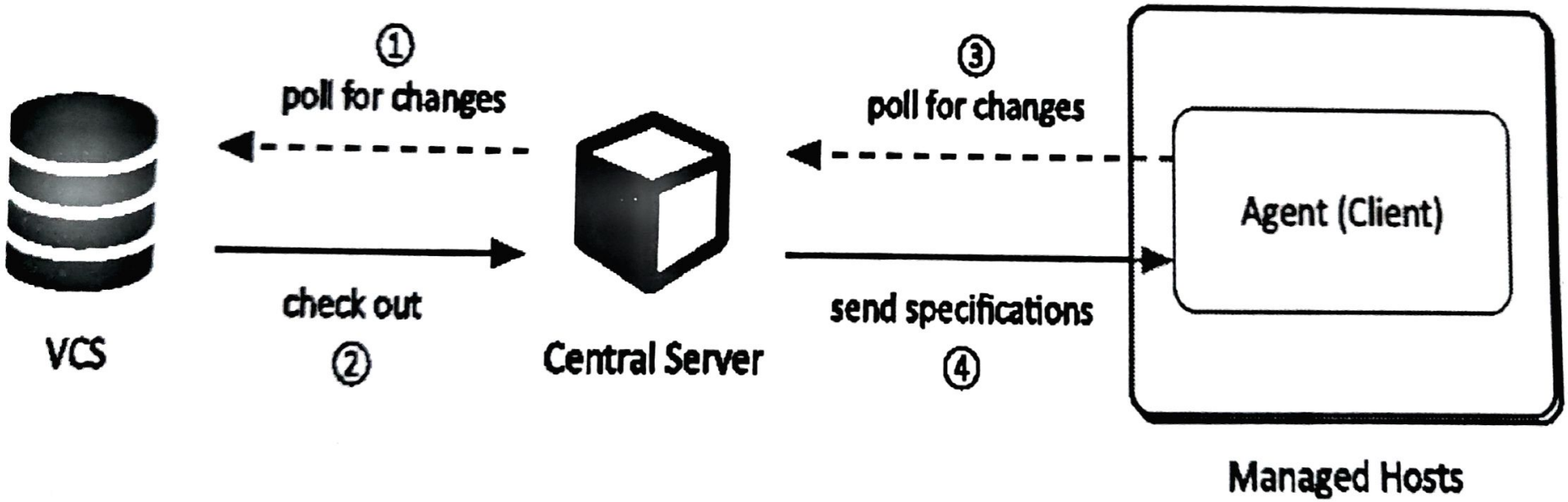
Ansible Architecture

- Taking control of your environment with a single tool has a numerous advantages.
- With Ansible, you can control server deployment configuration, making everything consistent. With modules and plugins, you can build or 'hook' into other applications and control them as well!

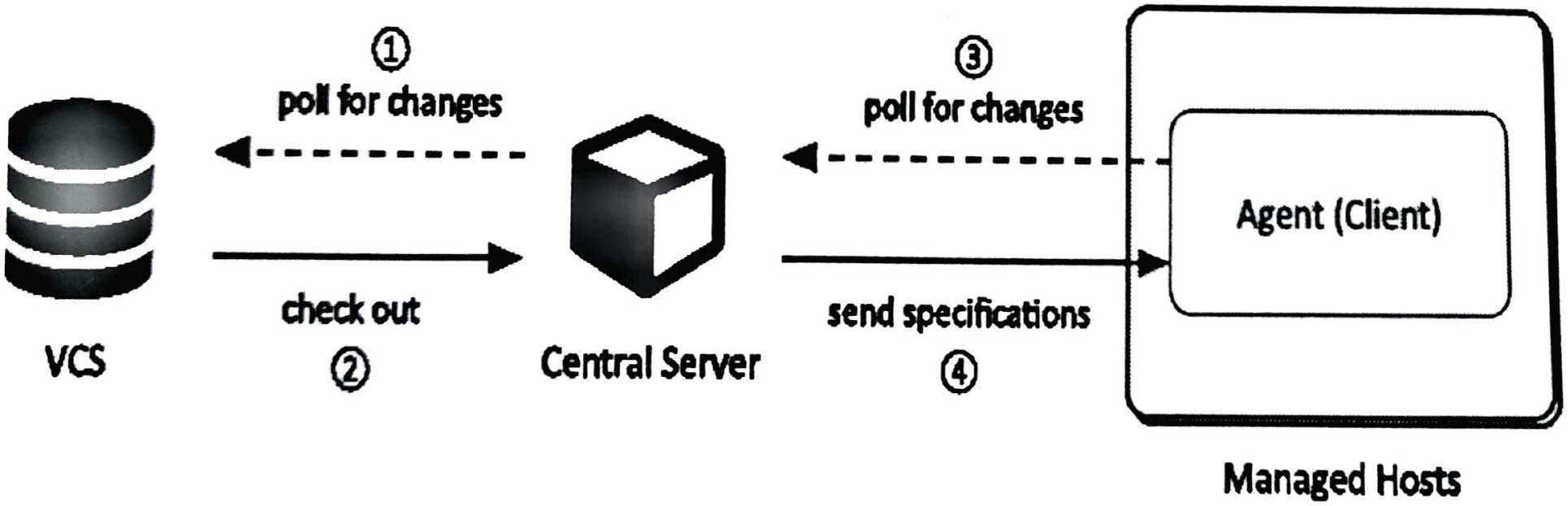
Agent-Based Architectures



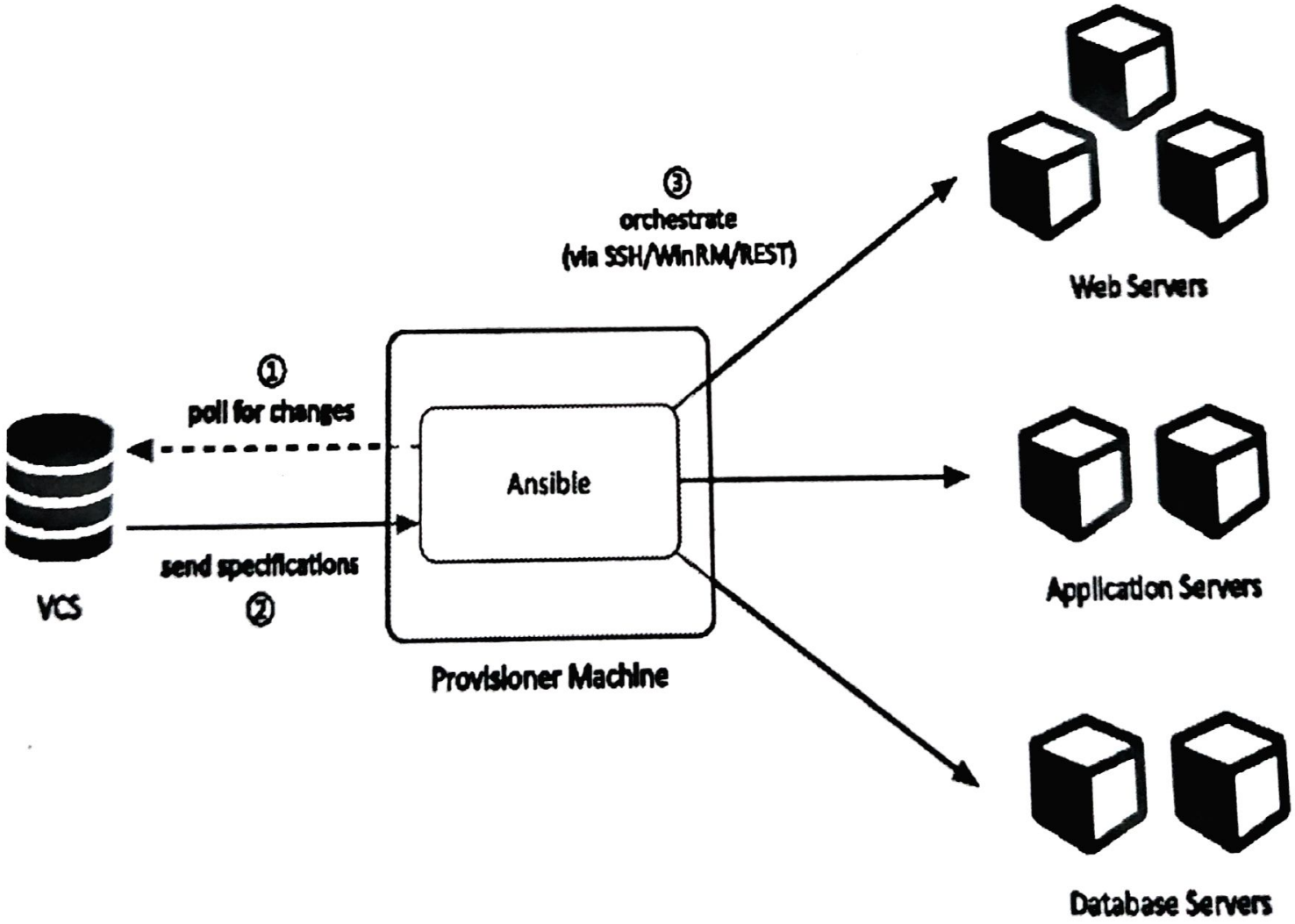
Agent-Based Architectures



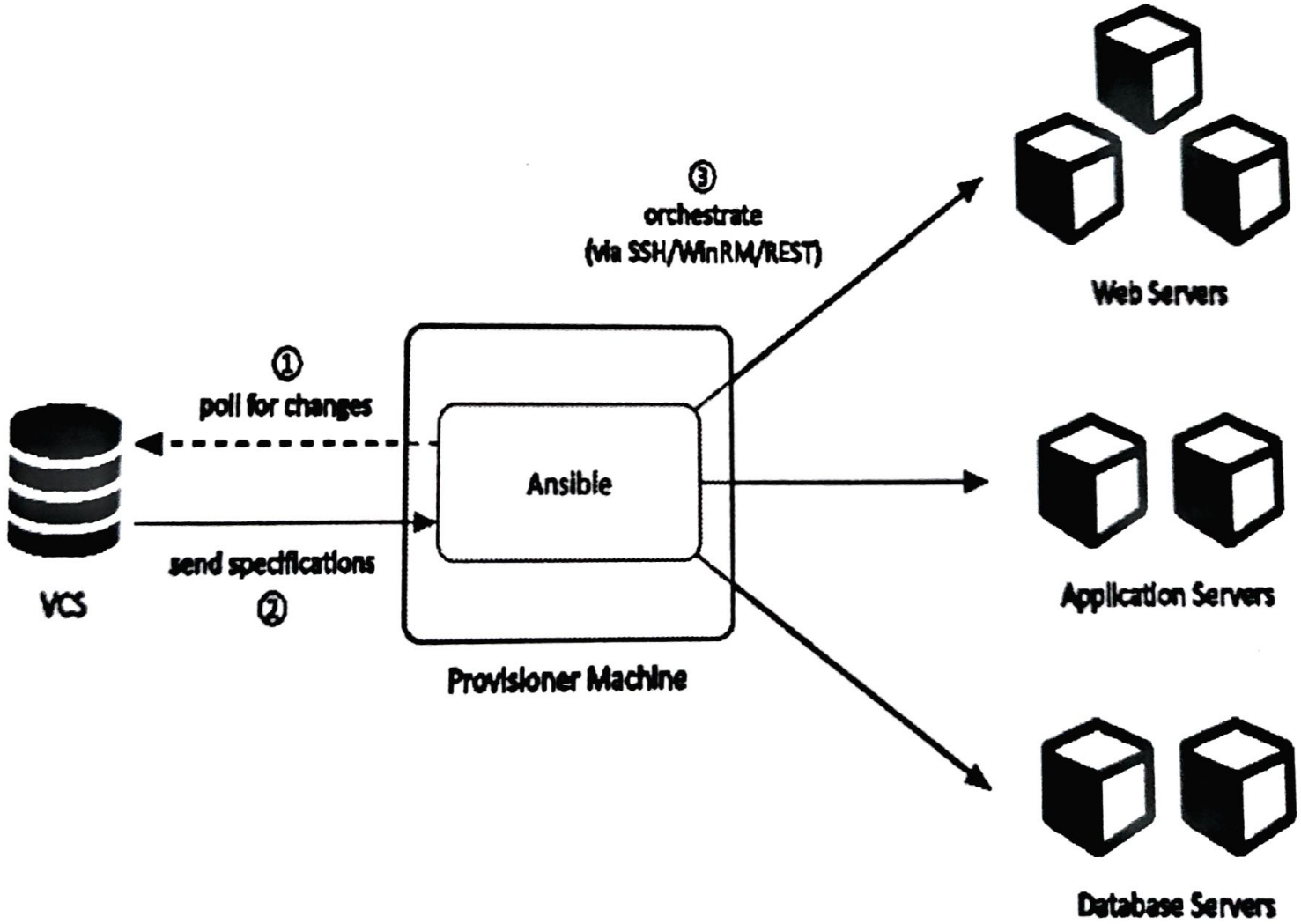
Agent-Based Architectures



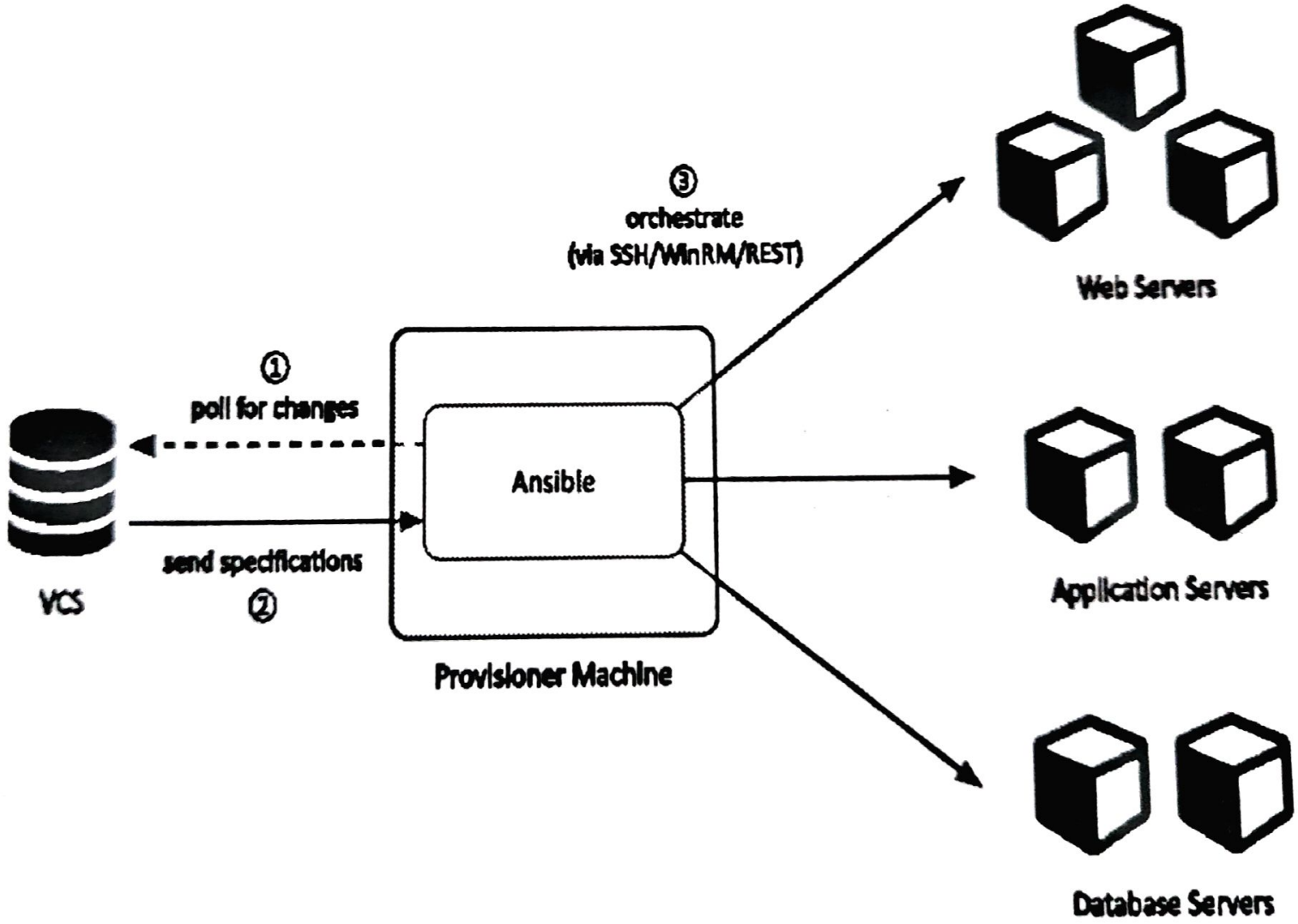
Ansible's Agentless Architecture



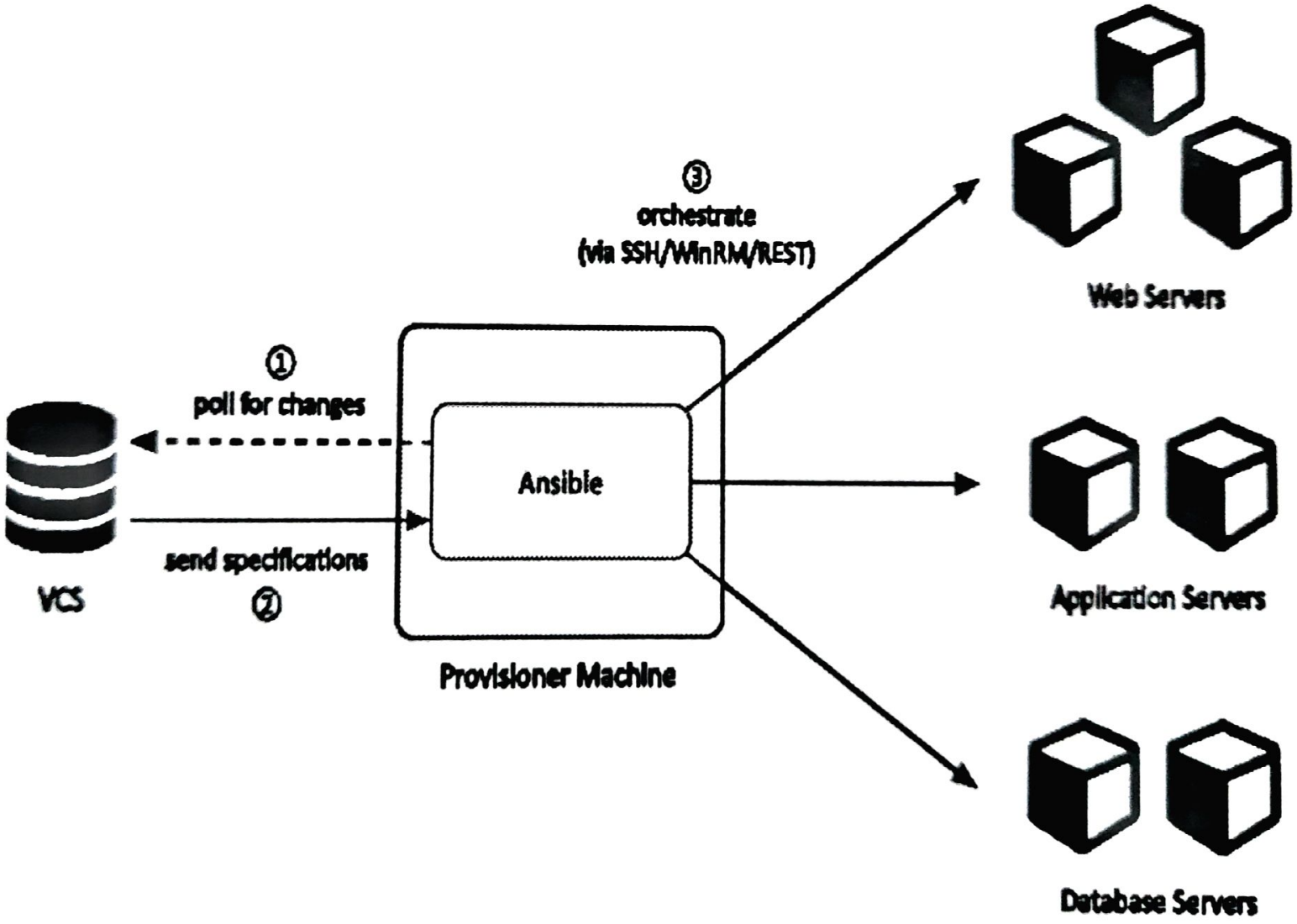
Ansible's Agentless Architecture



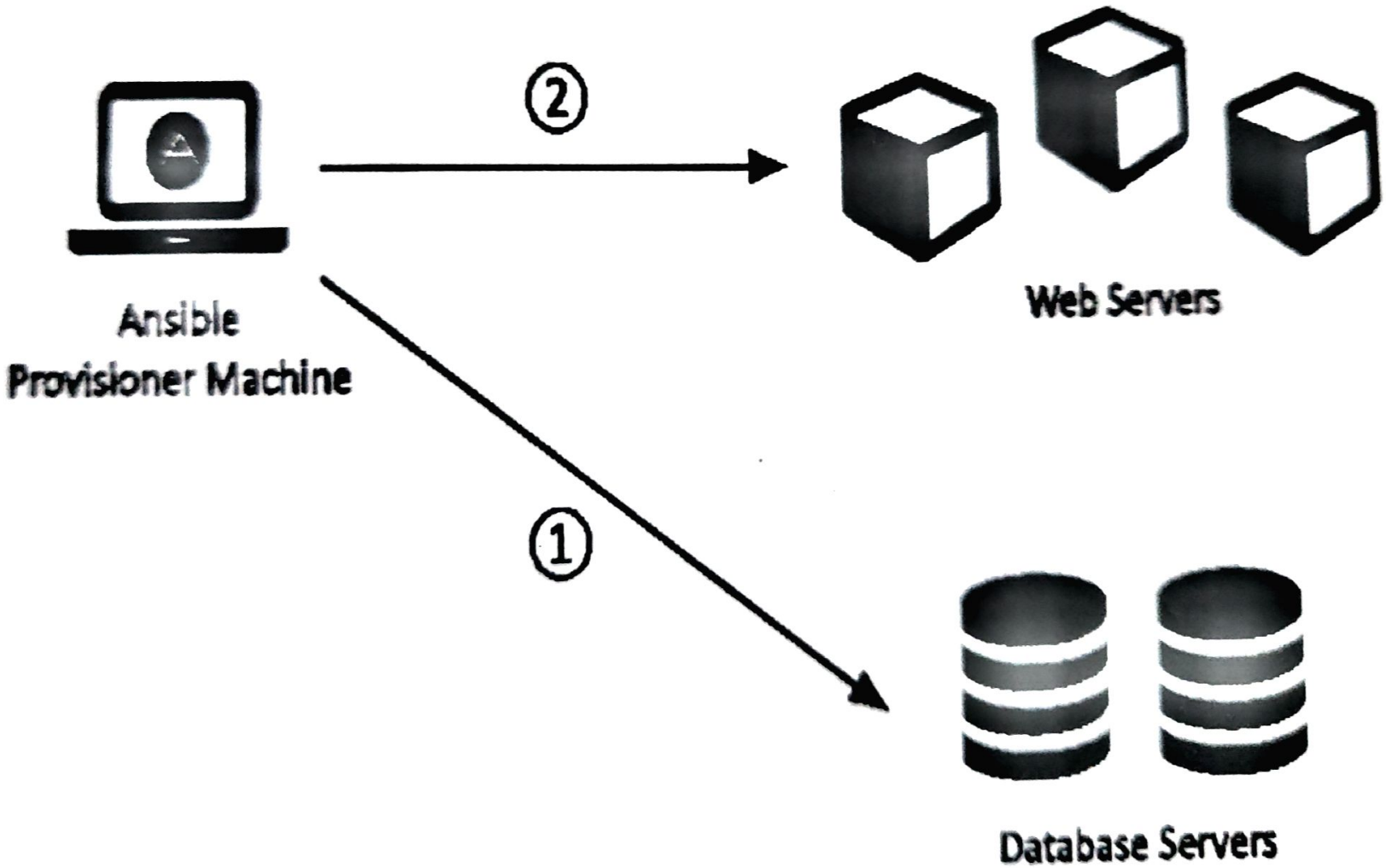
Ansible's Agentless Architecture



Ansible's Agentless Architecture



Ansible is an Orchestration Engine



Installation and Configuration

- Be sure to install '*epel-release' first and then update your caches (if CentOS).*

- Controll Server:
  - yum instll epel-release
  - yum update
  - yum install git python python-devel python-pip openssl ansible
  - ansible --version

User Accounts

- Create a user called '*ansible' on the server you intend to use Ansible to run playbooks from AND* each of the Ansible nodes you intend to run playbooks on. Set the user as a sudo-capable user and include the NOPASSWD: ALL directive in */etc/sudoers.*

- *-- Add ansible user (Repeat the same step for remaining systems)*
- *-- useradd ansible*
- *-- passwd ansible*
- *-- visudo (Add the following line in root section)*
- *ansible ALL=(ALL) NOPASSWD: ALL*

User Accounts – Password less login

- Create an SSH key with *ssh-keygen on the Ansible server. Exchange that key using ssh-copy-id on each* of the nodes you are running playbooks on. This allows the playbook to run with escalated privileges as needed.

- -- Password less login:
- -- login with ansible user
- -- run the 'ssh-keygen'
- -- run ssh-copy-id user@ip

Configuration Files

- /etc/ansible/ansible.cfg
    - Primary Ansible configuration file (agentless, daemon-less configuration, read on each ansible command run)
    - Uncomment "inventory" and “ sudo user" fields.
    - No restarts Required

    - vi /etc/ansible/ansible.conf
    - -- enable
    - sudo user:
    - inventory location:

Ansible Concepts: Inventories

Specify the environment Ansible operates in.

» Groups and hosts are defined in inventories

» Use inventories for staging and production

» Text files expressed in an INI-like format

Ansible Concepts: Inventories

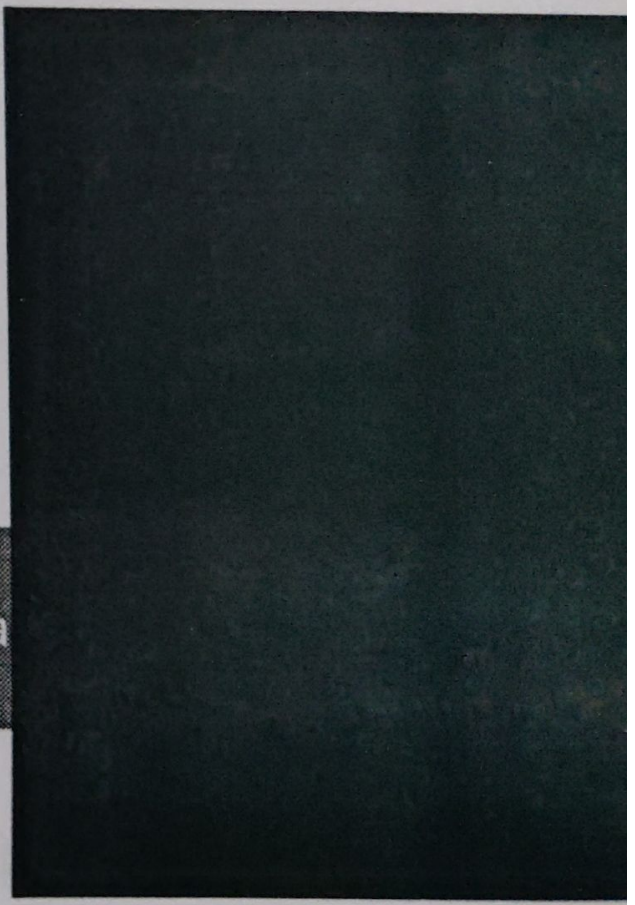
```
# production

[balancers]
www.example.com

[webservers]
www[0-9].example.com

[dbservers]
db[a:f].example.com

[monitoring]
dynatrace.example.com
```

Group

Host

Numeric Range

Alphabetic Ra



Ansible Concepts: Inventories

Customizing Hosts file (/etc/ansible/hosts)

-- [local]

ansible1.rnstech.com

[centos]

ansible2.rnstech.com

[ubuntu]

ansible3.rnstech.com

Running Arbitrary Commands

- Arbitrary commands can be run against hosts or groups of hosts at the command line, one at a time.
- for example: In order to list the contents of the home directory for the ansible user,
    - ansible GROUPNAME -a "ls -al /home/ansible“

- Running a command that requires sudo privileges should not be run with the sudo command, but rather the sudo parameter in the ansible command itself, like so:
    - ansible GROUPNAME -s -a "ls -al /var/log/messages"

Running Arbitrary Commands

- You can also execute a single module against one or more hosts at the command line by using the module parameter. As an example:
  - ansible GROUPNAME -s -m yum -a "name=httpd state=latest"

- Test if all machines in your inventory respond to a ping request:
  - ansible all -m ping

Ansible Concepts: Playbooks

*Defines **sequences** of tasks (Plays) to be executed on a group of hosts.*

» Describes **policies** machines under management shall enforce

» Contains **variables, tasks, handlers, files, templates** and **roles**

» Expressed in **YAML**

What is YAML?

- YAML stands for "**Yet Another Markup Language.**"
- In short, YAML is meant to be a "human-readable data serialization format."
- In other words, it's meant for non-computers (us) to be able to read easily and recognize the significance of its content without complex translation applications.
- It was designed to be easy to map to high-level languages, and you will often see it compared to key/value lists, associative arrays, and data outlines.

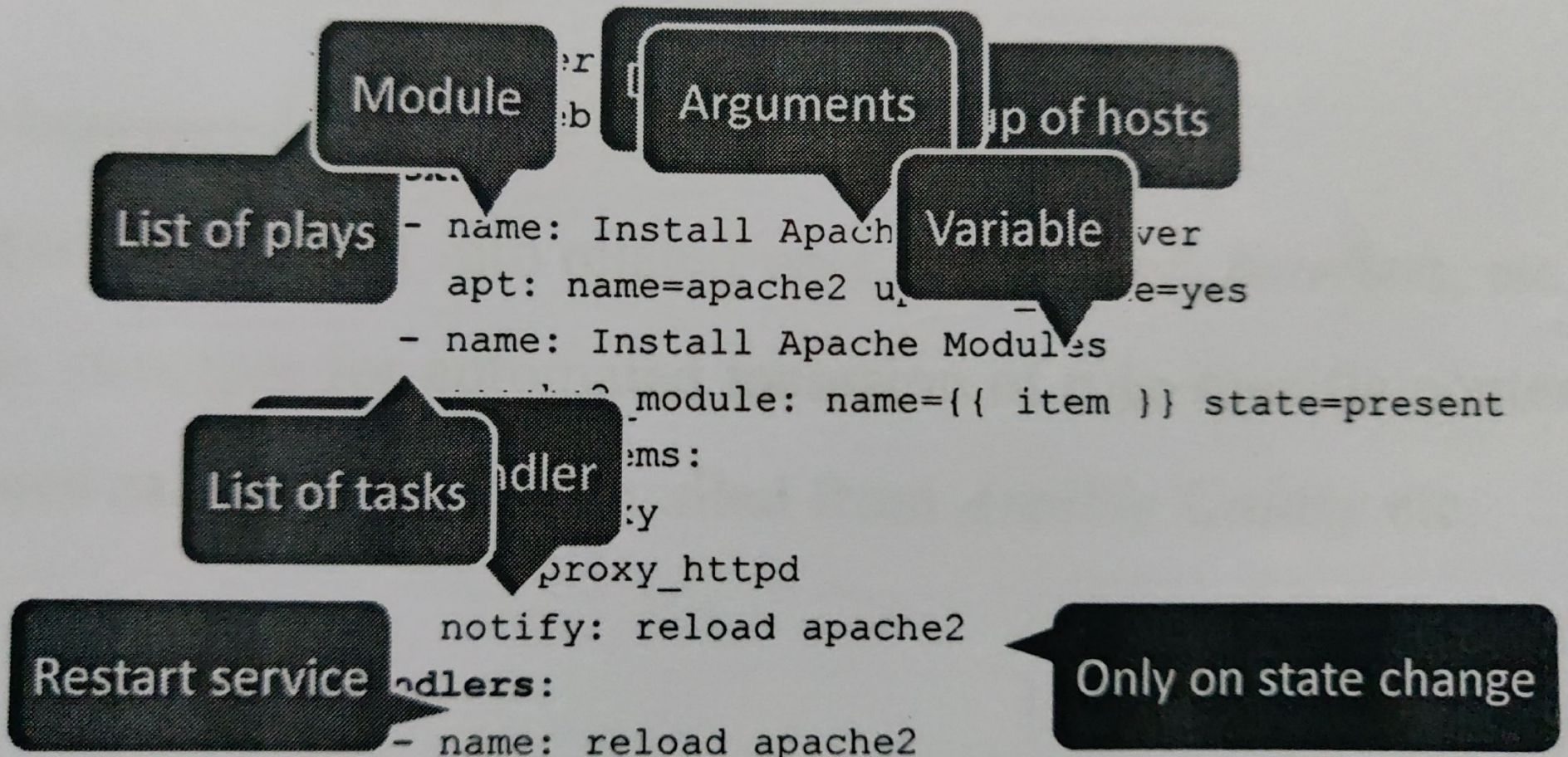
Structure

- As a data structure, YAML most closely resembles an outline or list of things with basic descriptions.
- For example, if we wanted to list our favorite movies in a way that YAML processing engines would be able to recognize, our YAML file would contain something like:
- --- # Our Favorite Movies of All Time
    - - The Terminator
    - - Star Trek
    - - Star Wars

Sample Playbook with Major Sections

- --- # COMMENT ABOUT PLAYBOOK
  - - hosts: hostsToRunAgainst
  - remote_user: ansible
  - become: yes
  - become_method: sudo
  - connection: ssh
  - gather_facts: no
  - vars:
    - var1: value
    - var2: value
  - tasks:
    - - name: Some description of what we are doing
    - yum:
      - name: httpd
      - state: latest
    - notify:
      - - startservice
  - handlers:
    - - name: startservice
    - service:
      - name: httpd
      - state: restarted

Ansible Concepts: Playbooks



Ansible Concepts: Templates

- Templates are used with variable substitution

- They are processed by the Jinja2 templating system
  - http://jinja.pocoo.org/docs/

- Useful for creating premade config files and then substituting the variables when the playbook runs

Ansible Concepts: Roles

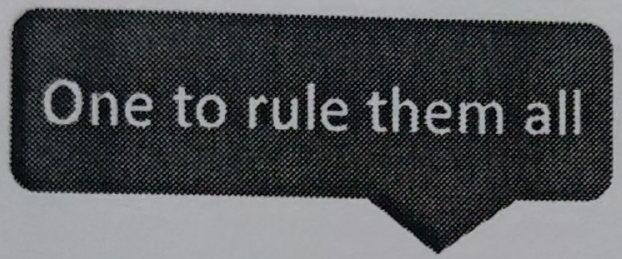
The best way to organize your playbooks.

» Structure content into related *vars*, *tasks*, *files*, *handlers*, etc.

» File structure for automated inclusion of role-specific content

» Roles can be shared and pulled from *Ansible Galaxy* etc.

Ansible Concepts: Playbooks




```
--- # playbook.yml
- include: dbservers.yml
- include: webservers.yml
- include: balancers.yml
- include: monitoring.yml
```

ansible-playbook

- Calling a playbook
- ansible-playbook /path/to/playbook.yaml