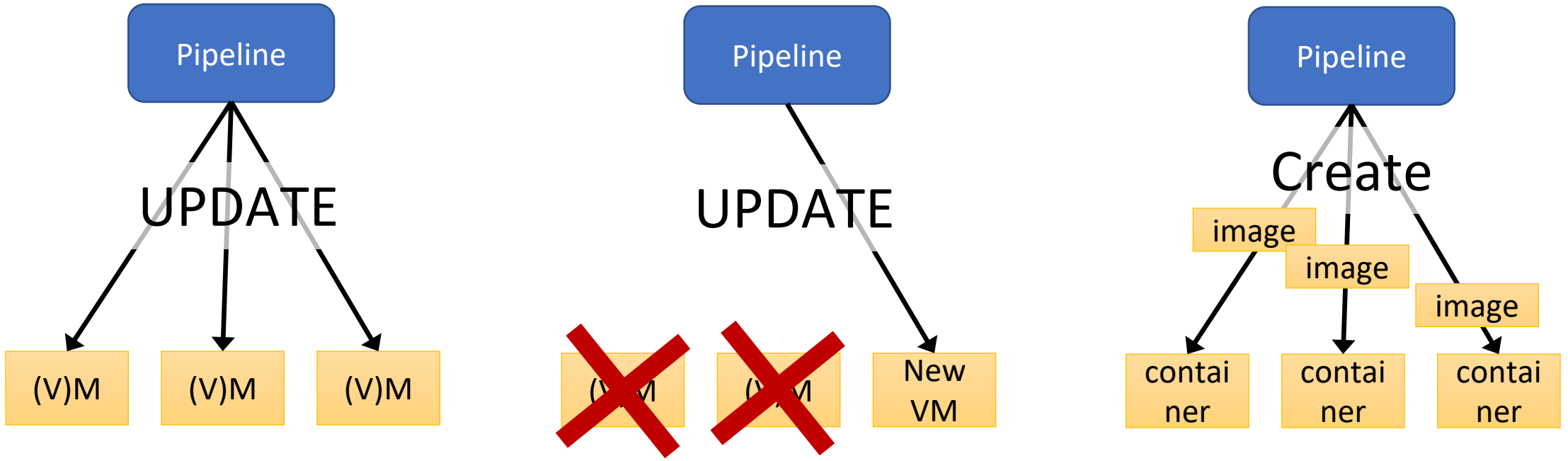
Intro to Ansible

Ansible exercise

Backgroud (slide from last week)



Alternative approaches for delivery

- Set-up everything when image is created
  - Very static
- Make the container to auto-update
  - You need to know in advance what might change
- ~~Put stuff to shared folder (use volume)~~
- Use configuration tools
  - Work also for full virtual machines and computers

Tampereen yliopisto
Tampere University

Ansible (https://www.ansible.com)

Automation engine for

- Provisioning
- Configuration Management
- App Deployment
- Continuous Delivery
- Security Automation
- Orchestration

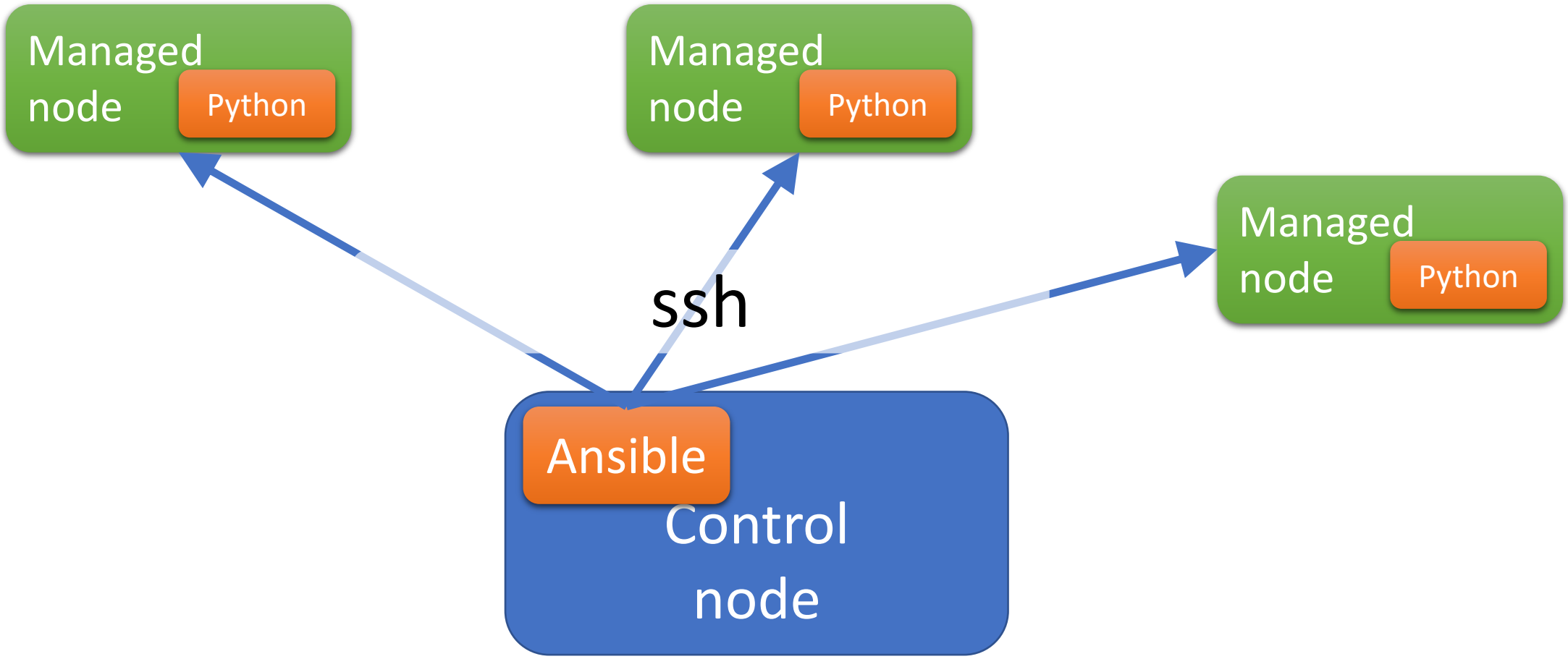
uses YAML, in the form of Ansible Playbooks

Tampereen yliopisto
Tampere University

Ansible

- Ansible works by connecting to your nodes and pushing out small programs, called "Ansible modules" to them.
- These programs are written to be resource models of the desired state of the system.
- Ansible then executes these modules (over SSH by default), and removes them when finished.
- Your library of modules can reside on any machine, and there are no servers, daemons, or databases required.
- Typically, you'll work with your favourite terminal program, a text editor, and probably a version control system to keep track of changes to your content.
- A short video:
  - https://www.ansible.com/resources/videos/quick-start-video

Architecture



Tampereen yliopisto
Tampere University

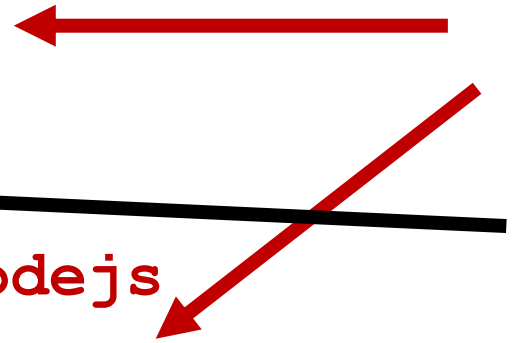
Docker containers as targets

- Since we do not enough virtual machines, lets use Docker images
- Complicates the exercise,
- but allows you to learn more about Docker

Tampereen yliopisto
Tampere University

Creating the docker image 1/2

```
FROM debian
USER root
# Copy application itself:
COPY . /home
WORKDIR /home
RUN apt-get update
RUN apt-get install -y nodejs
ENTRYPOINT node server.js
```



Debugging aid

Time consuming init

Docker build –t utest

Tampereen yliopisto
Tampere University

Creating the docker image 2/2

```
1       FROM  utest


2       RUN apt-get install -y openssh-server
3       RUN sed -i 's/PermitRootLogin prohibit-password/PermitRootLogin yes/'
        /etc/ssh/sshd_config
4       RUN apt-get install -y net-tools
5       RUN useradd -m -s /bin/bash -G sudo -p $(openssl passwd -1 eee) ssluser


6       RUN apt-get install -y python3
7       RUN apt-get install -y sudo


8       ENV PORT=8894
9       EXPOSE 22


10      ENTRYPOINT service ssh start && node server.js
```

SSH support two alternative ways for authentication

Password

- Used in the previous slide
- Not very secure
- You can use, but gives at most 80% of the maximum points

Public/private keypair

- Public key of your computer is installed to the host
- More secure
- If you want 100% of maximum points, you should use this (building of the image need to be changed)

Info

- Short:https://unix.stackexchange.com/questions/210228/add-a-user-without-password-but-with-ssh-and-public-key
- Long: https://www.ssh.com/academy/ssh/key

Example ansible playbook

```
---
- hosts: webservers
  vars:
     http_port: 80
     max_clients: 200
  remote_user: root
  tasks:
  - name: ensure apache is at the
          latest version
    yum:
      name: httpd
      state: latest
  - name: write the apache config
          file.
    template:
      src: /srv/httpd.j2
      dest:/etc/httpd.conf
    notify:
      - restart apache
  - name: ensure apache is
          running
    service:
      name: httpd
      state: started

handlers:
    - name: restart apache
      service:
        name: httpd
        state: restarted
```

Tampereen yliopisto
Tampere University

Apt vs yum

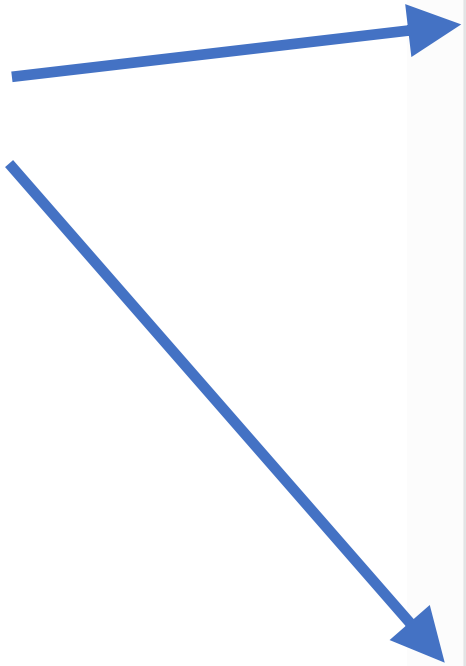
Operating System	Format	Tool(s)
Debian	.deb	apt, apt-cache, apt-get, dpkg
Ubuntu	.deb	apt, apt-cache, apt-get, dpkg
CentOS	.rpm	yum
Fedora	.rpm	dnf
FreeBSD	Ports, .txz	make, pkg

Tampereen yliopisto
Tampere University

Sidenode apt vs yum examples

Task	apt (deb)	yum (rpm)	zypper (rpm)
Install from repository	apt-get install pkg-name	yum install pkg-name	zypper install pkg-name
Update package	apt-get install pkg-name	yum update pkg-name	zypper update -t package pkg-name
Remove package	apt-get remove pkg-name	yum erase pkg-name	zypper remove pkg-name
Install from package file	dpkg -i pkg-name	yum localinstall pkg-name	zypper install pkg-name

There can be multiple plays



```yaml
---
- hosts: webservers
  remote_user: root

  tasks:
  - name: ensure apache is at the latest version
    yum:
      name: httpd
      state: latest
  - name: write the apache config file
    template:
      src: /srv/httpd.j2
      dest: /etc/httpd.conf

- hosts: databases
  remote_user: root

  tasks:
  - name: ensure postgresql is at the latest version
    yum:
      name: postgresql
      state: latest
  - name: ensure that postgresql is started
    service:
      name: postgresql
      state: started
```

Tampereen yliopisto
Tampere University

The exercise in short

- Read Ansible tutorial to understand how it works. A good starting point is: https://docs.ansible.com/ansible/latest/user_guide/intro_getting_started.html
- Prepare a docker image that can be used as a target. See details in below.
- Install Ansible in your computer.
- Make simple playbook
  - Check that the image has the latest version of git version management system
  - Queries the uptime (Linux command uptime) of target host
- Return result to Plus

Tampereen yliopisto
Tampere University

Testing your Ansible

1. Start one container from the image, get its IP-address.
   (in case of password-based authentication you need a manual login after start)
2. Ensure that the IP address is in /etc/ansible/hosts
3. Run the playbook
4. Copy the output (O1)
5. Run the playbook again
6. Copy that output, too (O2)
7. Start a second contained from the image, get its IP-address.
8. Ensure that this IP address is in /etc/ansible/hosts, too.
9. Run the playbook
10. Copy the output (O3)
11. Run the playbook again
12. Copy that output, too (O4)

Tampereen yliopisto
Tampere University

Submission

- Git link of the code (teacher may want do git clone). Use different folder or subrepo from earlier exercises.
- The report should have a “report.pdf” with the following contents.
  - All the copied output (O1,O2,O3,O4)
  - Comments on what was easy and what was difficult.

Tampereen yliopisto
Tampere University

Main principles (https://continuousdelivery.com/principles/)

- Build quality in
- Work in small batches
- Computers perform repetitive tasks, people solve problems
- Relentlessly pursue continuous improvement
- Everyone is responsible

Sound familiar from somewhere?

Tampereen yliopisto
Tampere University

CI – essential practices
(according to Humbley and Farley)

- Don’t check in on a broken code
- Always run all commits tests locally before committing, or get your CI server to do it for you
- Wait for commit tests to pass before moving on
- Never go home on a broken build
- Always be prepared to revert to the previous revisions
- Time-box fixing before reverting
- Don’t comment out failing tests
- Take responsible for all breakages that result from your changes
- Test-driven development

Tampereen yliopisto
Tampere University

Deployment essential pract.
(according to Humbley and Farley)

- Only build your binaries once
- Deploy the same way to every environment
- Smoke-test your deployments
- Deploy to copy of production
- Each change should propagate through the pipeline instantly
- If any part of pipeline fails, stop the line