Automating PMP Maintenance

Szymon Trocha (PSNC), Nikola Gacesa (AMRES)

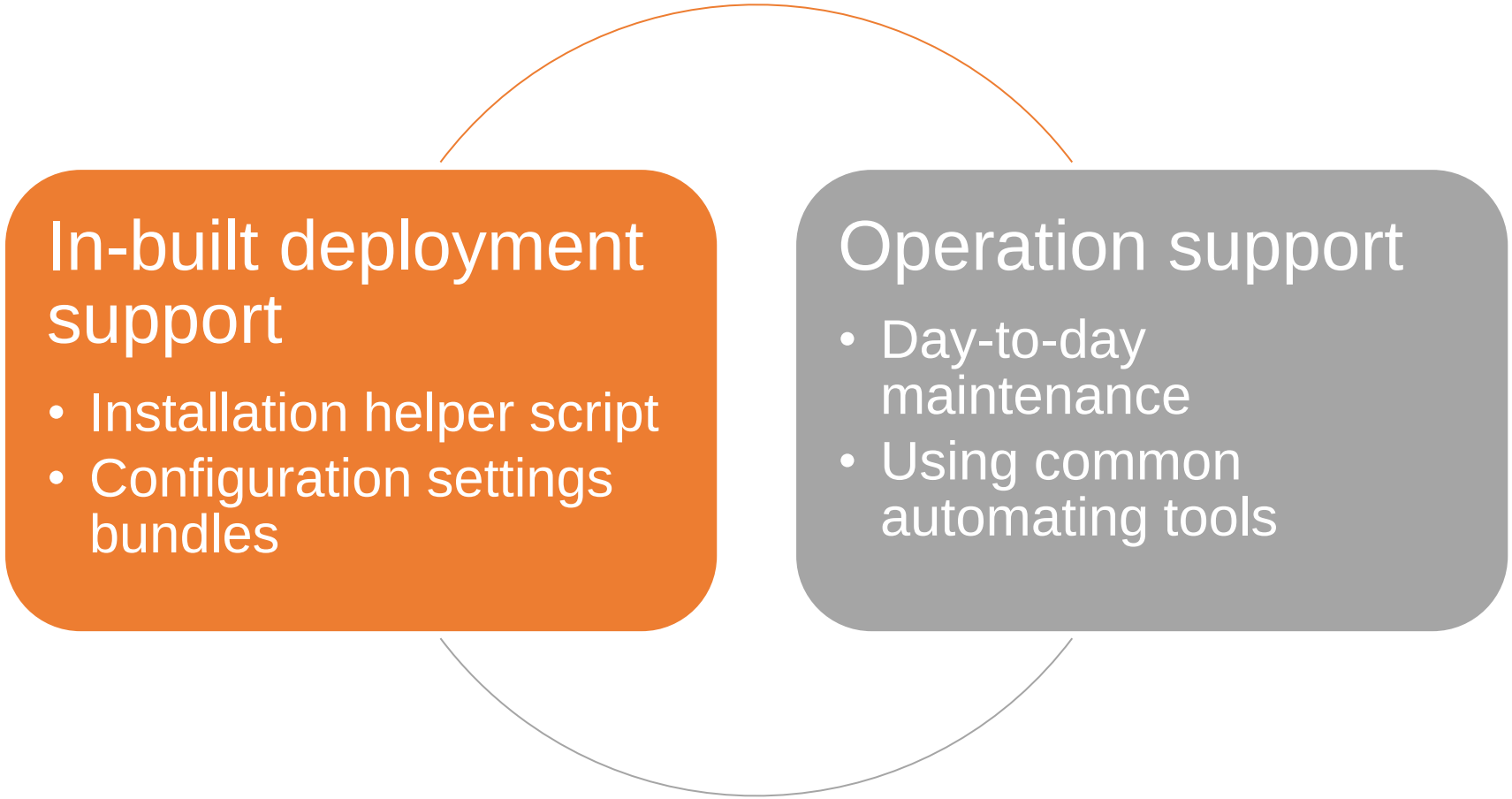
Network Performance and Monitoring Workshop, Prague, Czech Republic

1-2 April 2025

Why do we automate?

- To speed-up deployment process
- To improve efficiency for repetitive tasks
- To increase productivity giving team members extra time for other tasks
- To provide common tools to operate the service
- To reduce human errors
- To transfer knowledge

Pilars of perfSONAR automation



In-built deployment support

Installation automation script

- Automatically performs the first two steps of installation (repo config + bundle installation)
- `# curl -s https://downloads.perfsonar.net/install | sh -s - toolkit`

Configuration settings bundles

- Optional add-on packages
  - `perfsonar-toolkit-ntp` - Automatically detects closest NTP servers and sets them in ntp.conf
  - `perfsonar-toolkit-security` - Adds default firewall rules and installs fail2ban
  - `perfsonar-toolkit-servicewatcher` - Adds a cron job that checks if services are still running
  - `perfsonar-toolkit-sysctl` - Adds default sysctl tuning settings
  - `perfsonar-toolkit-systemenv-testpoint` - Configures auto-update and set some default logging locations

Operation supporting components

- Use common automating tools of your choice

- Ansible
- Bash scripts

What do we automate?

New host onboarding

Access (ssh keys, users, permissions)

Configuration (schdule, archiving)

Service control

Package Management

Updates

Automation examples (1/2)

Access configuration

- Hardening remote by limiting it only to specific team members with ssh keys
- For a newly deployed PMP node (testpoint or toolkit already installed)
- Requires ssh access with username/password and sudo privileges

Joining the mesh configuration

- In order for the node to become part of the mesh, we push a psconfig remote configuration
- Forcing refresh

Atuomation examples (2/2)

Archiving configuration

- In order for the node to publish test results to the archive, we trigger an archiving configuration

System services maintenance

- Service restart for the changes to take full effect or to recover
- Package management (install, update)

Playbooks (all over time)

```
root@pmp-archive:/etc/ansible/playbooks# ls
archiving.yml          keys.yml                    psconfig.yml
auto_updates.yml       maintenance.yml             ps-repo.yml
check-version.yml      network-settings.yml        remove-remote.yml
corrections.yml        node_exporter.yml           restart-scheduler.yml
cpugovernor.yml        pmp-archive-old.yml         setup.yml
files                  pmp-archive.yml             sysmon.yml
hosts-file.yml         privacy-notice.yml          users.yml
HOWTO.md               pscfg-json-change.yaml
root@pmp-archive:/etc/ansible/playbooks#
```

Example user management playbook items

- Check GÉANT team user access is set as intended
- Create a pssudo group
- Create a geantadmin user
- Set authorized keys
- Add useful user configuration files
- Make sure system updates are installed
- Install required and very useful packages - DEB
- Create a psudouser user
- Create a pswebadmin user
- Disable root access
- sshd restart

Automation examples - users and ssh keys

```
# ansible-playbook -kK playbooks/users.yml -e
"ansible_user=geantadmin" -l "{target_node_fqdn}"
```

```
…
- "files/authorized_keys/antoine"
        - "files/authorized_keys/dragan"
        - "files/authorized_keys/ljubomir"
        - "files/authorized_keys/nikola"
        - "files/authorized_keys/szymon"
      tags: [ 'nodes::geantadmin' ]
…
```

Automation examples - psconfig push with schedule

```
# ansible-playbook -kK playbooks/pscfg-json-change.yaml
-e "ansible_user=geantadmin" -l "{target_node_fqdn}"
```

```
…
- name: Add new remote configuration
      shell: |
        psconfig remote --configure-archives add
"https://pmp-archive.geant.org/psconfig/pscfg-pmp.json"
      register: add_result
…
```

Automation examples - remote archive configuration

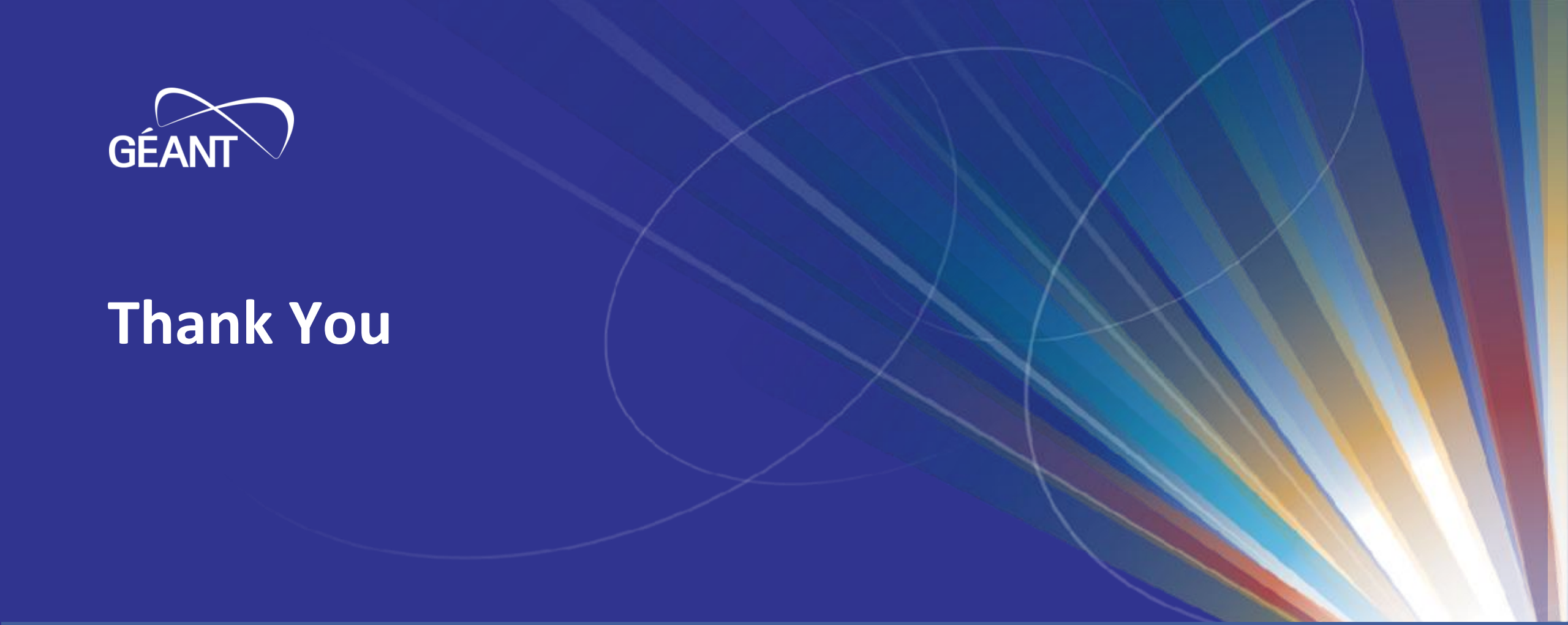
```
# ansible-playbook -kK playbooks/pmp-archive.yml -e
"ansible_user=geantadmin" -l "{target_node_fqdn}"
…
                "archiver": "http",
                "data": {
                    "schema": 3,
                    "_url": "https://{% raw %} scheduled_by_address
{% endraw %}/logstash",
                    "op": "put",
                    "_headers": {
                        "x-ps-observer": "{% raw %}
scheduled_by_address {% endraw %}",
                        "content-type": "application/json"
…
```

Automation examples - local service control

/home/geantadmin/restart_services - currently as a local BASH script on all nodes:

```
…
services=(
    "pscheduler-scheduler"
    "pscheduler-runner"
    "pscheduler-archiver"
    "pscheduler-ticker"
    "psconfig-pscheduler-agent"
    "owamp-server"
    "perfsonar-lsregistrationdaemon"
)

# Loop through each service and restart it
for service in "${services[@]}"; do
    echo "Attempting to restart $service..."
    systemctl restart "$service"
…
```



Thank You

Automating PMP Maintenance

Szymon Trocha (PSNC), Nikola Gacesa (AMRES)

Network Performance and Monitoring Workshop, Prague, Czech Republic

1-2 April 2025

The scientific work is published for the realization of the international project co-financed by Polish Ministry of Science and Higher Education from financial resources of the programme entitled "PMW"