Red Hat
Enterprise Linux

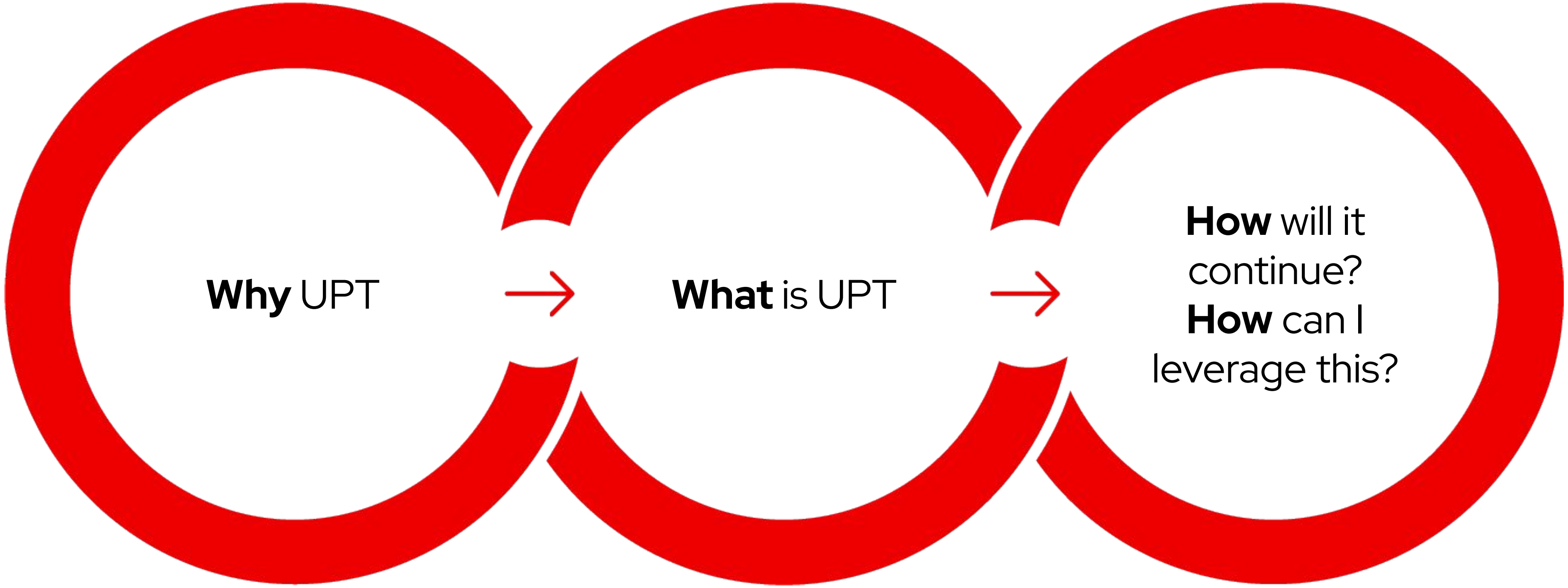
UPt! Your Provisioning of Linux Machines

Jakub Raček

Software Engineer @ Red Hat

Red Hat

UPT



Why UPT

→

What is UPT

→

How will it continue?
How can I leverage this?

“**Why?**” (a disclaimer)

- **Testing for CKI need to…**
  - Target Different Providers/Clouds (**UPT**)
    - No 1 project will do *all* the work for you
  - Run Kernel Testing (**UPT/restraint runner**)
  - Generate results
  - Have CI Compatible Output (**KCIDB**) -> reports / dashboards
  - Scale? (Gitlab pipelines)
  - Handle Transitioning & Legacy! (**legacy** workflow)
  - Avoid creating a new standard (**?**)
  - Be reliable (existing tests and test and tests)

“**What** is UPT?”

- Gitlab repo: https://gitlab.com/cki-project/upt
- Python 3 code
- **Provisioner** (mostly synchronous)
  - Beaker
  - AWS
  - 1minutetip (fast!)
  - (more coming soon?)
- **Test runner**
  - Run tests on any above

“What is provisioning?”

```
$ upt provision

UPT CLI started
Provisioning resources...
Waiting for 1 to be ready...
Provisioning confirmed for
resource_group #1
Resource(s) provisioned.

$ echo “Now we can ssh into it!”
```

“**What** does UPT eat?”

- **XML**: legacy workflow support
- **Yaml**: describe provisioning
- You can sort of edit it easier :-)
- Demo

“**What** does provisioner look like?”

“**What** is a test runner?”

```
restraint test runner started
Results for 10.0.139.128 will be present in the job artifacts in:
/home/jracek/src/cki-project/upt/run.done.07/results_0001/
Please see the index.html file in this directory.
Runner waiting for processes to finish...
* Running "restraint --conn-retries 15 --job /tmp/tmpgc5q875m/job.xml --rsh "ssh -o
StrictHostKeyChecking=no -o GSSAPIAuthentication=no -i /usr/share/qa-tools/1minutetip/1minutetip" --host
1=root@10.0.139.128 "...
Recipe: #1: Host 10.0.139.128: Running: /distribution/command
Recipe: #1: Subtask /distribution/command: FAIL Score: 1
Recipe: #1: Host 10.0.139.128: Completed: FAIL: /distribution/command
Recipe: #1: Completed: FAIL: /distribution/command
(A non-waived task failed. Kernel testing failed. Continuing with other tests.)
```

“**What** is the summary of this?”

- **Features**
  - A couple of provisioners (wrappers)
  - Incremental test results
  - CI system compatibility
  - Codified result evaluation
  - Re-running of tests on conditions
  - Fast prototyping
  - 100% unit test coverage

“**How** can I leverage this?”

- **Target provisioners / providers / clouds**
  - Wouldn’t it be nice to target any provisioner?
- **Target resources to run tests**
  - How can I get/send **results** from a **lab**?
  - Simple, repeatable workflow
  - **Kernel testing** and **KCIDB**

“**How** will it continue?”

- DataWarehouse
- Translating hardware requirements into provisioner-specific terms (tmt?)
- Transition layers
- ~~Better AWS setup?~~
- Better re-running
- Better docs :-)
- Better community traction :-)

Q & A

Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.

in linkedin.com/company/red-hat

▶ youtube.com/user/RedHatVideos

f facebook.com/redhatinc

twitter.com/RedHat