WEBINAR

F5 - Advanced Automation

25.10.2023 | 9:30-11:00

SOITRON*

AGENDA

- Presentation
  - Part 1: Intro - Motivation, Mission, Decision
  - Part 2: Soitron Automation Tool
  - Part 3: Live demo
  - Part 4: Some tips for beginners
- Discussion and summary
- You can ask direct or write questions at Slido
- Presentation is recorded

Join at
slido.com
#soitron
passcode: tr4kxi

Challenge

A few years ago...

Number of services

DC: ~250 services, 3x geo-datacenter
(DMZ: ~150 services, 2x geo-datacenter)

Migrate data center
from “**legacy**” to “**software-defined**”

“Legacy” technologies:
* Cisco SW/RT, ASA/FWSM(!)
* Cisco ACE(!!!), f5 BIG-IP
* Cisco GSS(!!) (“geo-dns”)
* 10Gbps
* CLI based configuration
* uptime 10y+, EoS, EoL...
* No/limited API

“SD-DC” technologies:
* Cisco ACI (Nexus)
* Cisco FTD/FMC
* f5 Viprion
(LTM, AFM, DNS, WAF)
* 40/100Gbps
* Containers
* Open API

Mission

Service migration
+
Change of mind

Primary goal

Service migration
with full-automation

Secondary goal

Change
“Legacy thinking”
to “NetOps thinking”

Why?
* Migration accelerating
* Reducing human error (simple inputs + automation)
* Uniform configuration style
* Move to NetOps/DevOps thinking

Decision

Part 1/2

There is **no perfect** automation tool!

ANSIBLE

puppet

POSTMAN

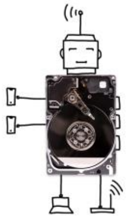
NSO

CISCO ACI

HashiCorp Terraform

CHEF

SALTSTACK



Decision

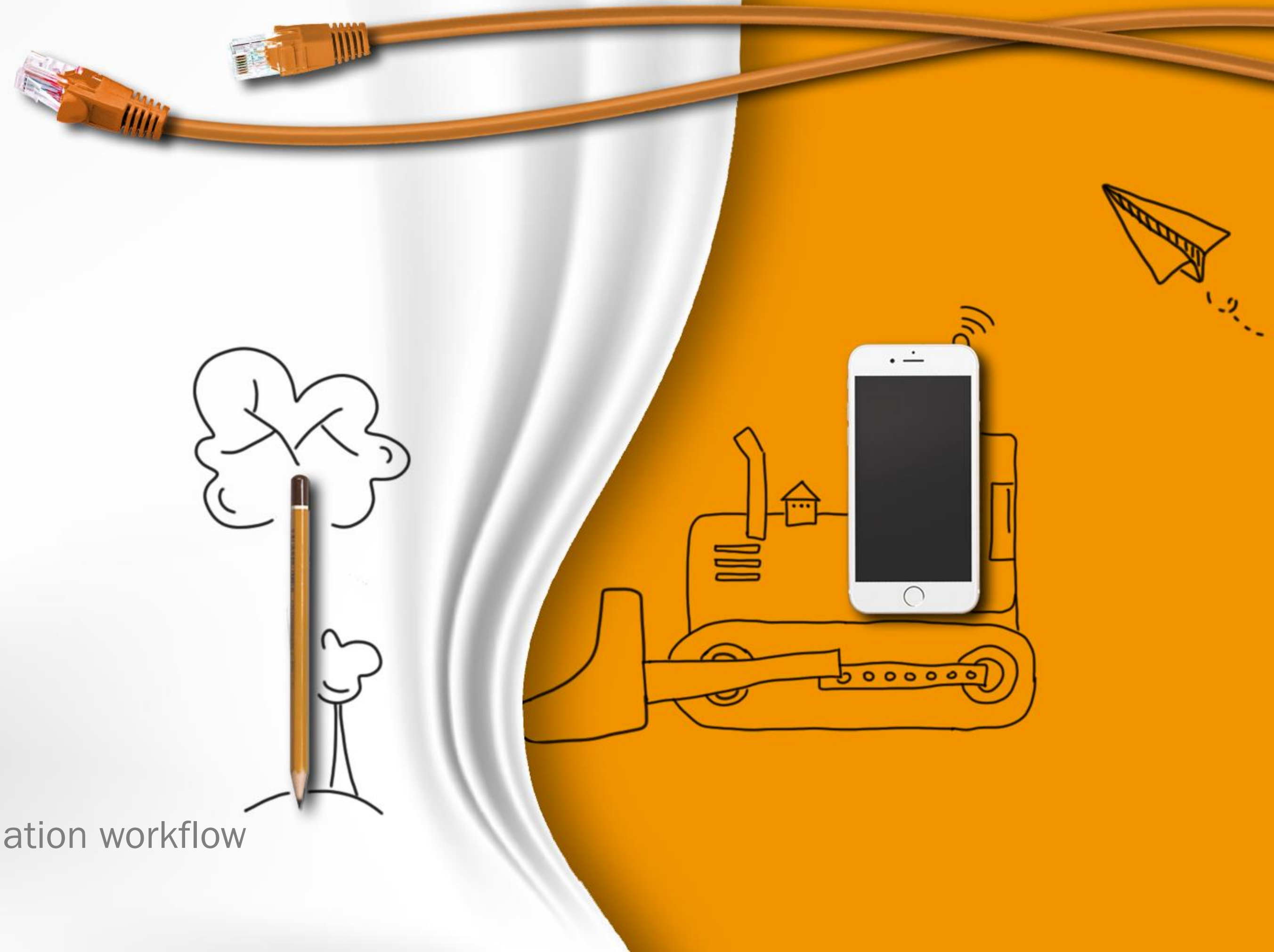
Part 2/2

Our custom on-premise solution based on…

ANSIBLE

python™

GitLab



Our Solution

(Not only) For BIG-IP automation workflow

F5 Solution

In short...

Focus to...

- The same workflow regardless end device
- Configuration/migration accelerating
- Reducing human error
- Configuration without expert level knowledge

F5 Automation...

- First version:
  - Preferred IMPERATIVE because Declarative had some limitations (~2019-2020)
  - Non-atomic (imperative)
- Today (2022+):
  - All configuration is DECLARATIVE (expect onboarding)
  - Almost ideal scenario!
  - Ready for the future, ready for the BIG-IP Next

F5 integration – (hi)story

History (~2015-2019)

- iWorkflow
- BIG-IQ cloud and orchestration
- iApps

Today (~2019+)

- F5 BIG-IP Application Services 3 Extension (AS3)
- F5 BIG-IP Application Services Templates (FAST)
- F5 BIG-IP Declarative Onboarding (DO)
- F5 BIG-IP Telemetry Streaming (TS)
- F5 BIG-IP WAF Declarative Policy
- F5 BIG-IP Automation Config Converter (ACC)
- F5 SDK (Python)
- ...and many more (https://clouddocs.f5.com/)

F5 resources

Ansible Collections

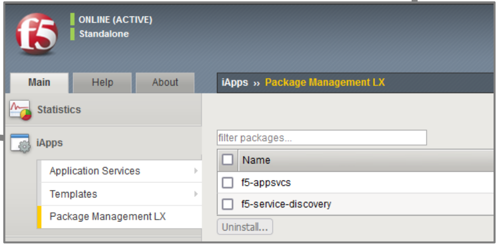
- Imperative (f5_modules) https://galaxy.ansible.com/ui/repo/published/f5networks/f5_modules/
- Declarative (f5_bigip) https://galaxy.ansible.com/ui/repo/published/f5networks/f5_bigip/
- F5OS (r-series, velos) https://galaxy.ansible.com/ui/repo/published/f5networks/f5os/

BIG-IP Extensions documentation

- BIG-IP Application Services 3 doc https://clouddocs.f5.com/products/extensions/f5-appsvcs-extension/latest/userguide/
- BIG-IP Declarative Onboarding doc https://clouddocs.f5.com/products/extensions/f5-declarative-onboarding/latest/

F5 appsvsc extension

- RPM package https://github.com/F5Networks/f5-appsvcs-extension
- + Postman collection
- + JSON schema



Join at

slido.com

#soitron

Passcode: **tr4kxi**

Our Solution

Not for BIG-IP only

SOITRONAUT

Soitron Automation Tool

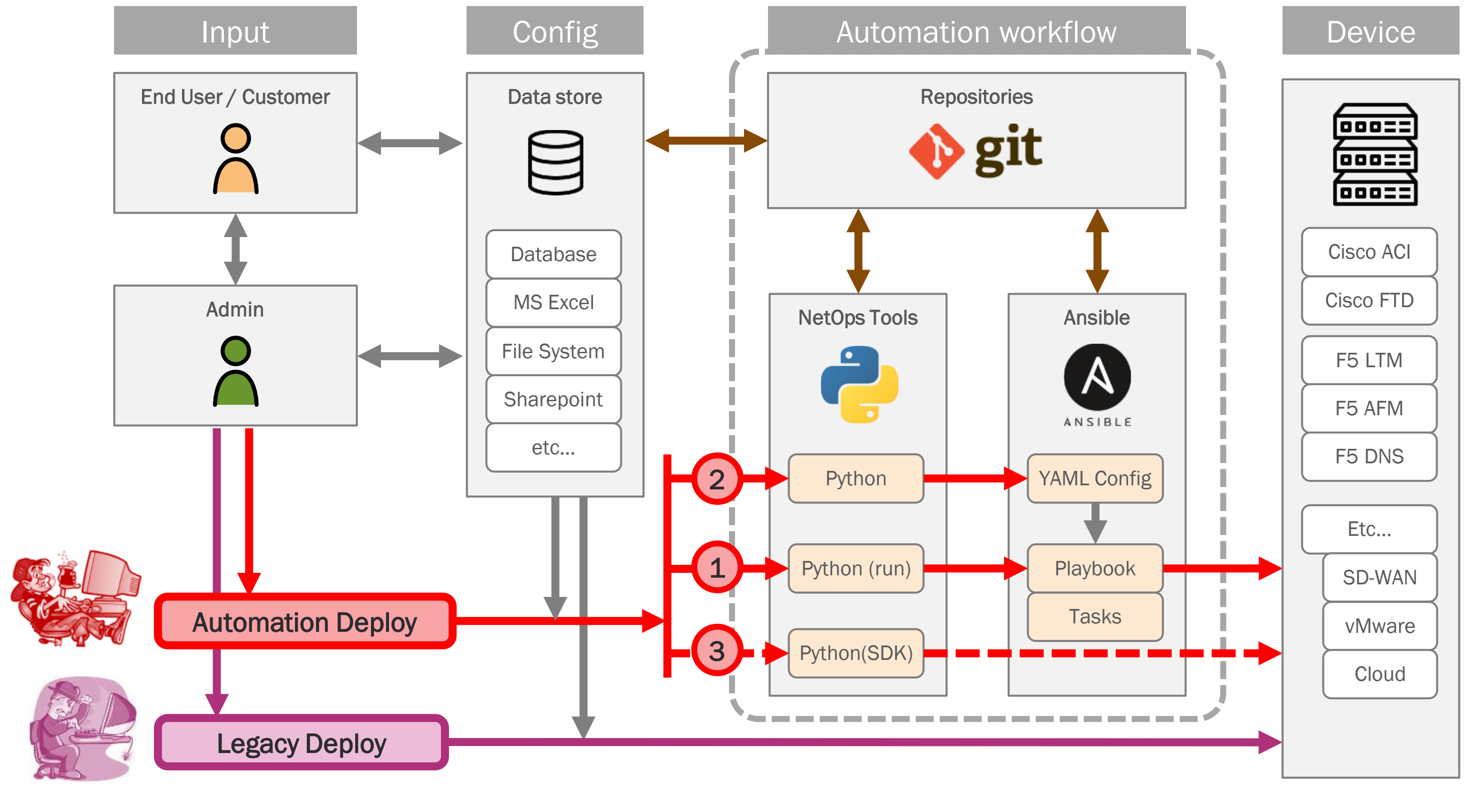
Custom automation tool(s)

(not for BIG-IP only)

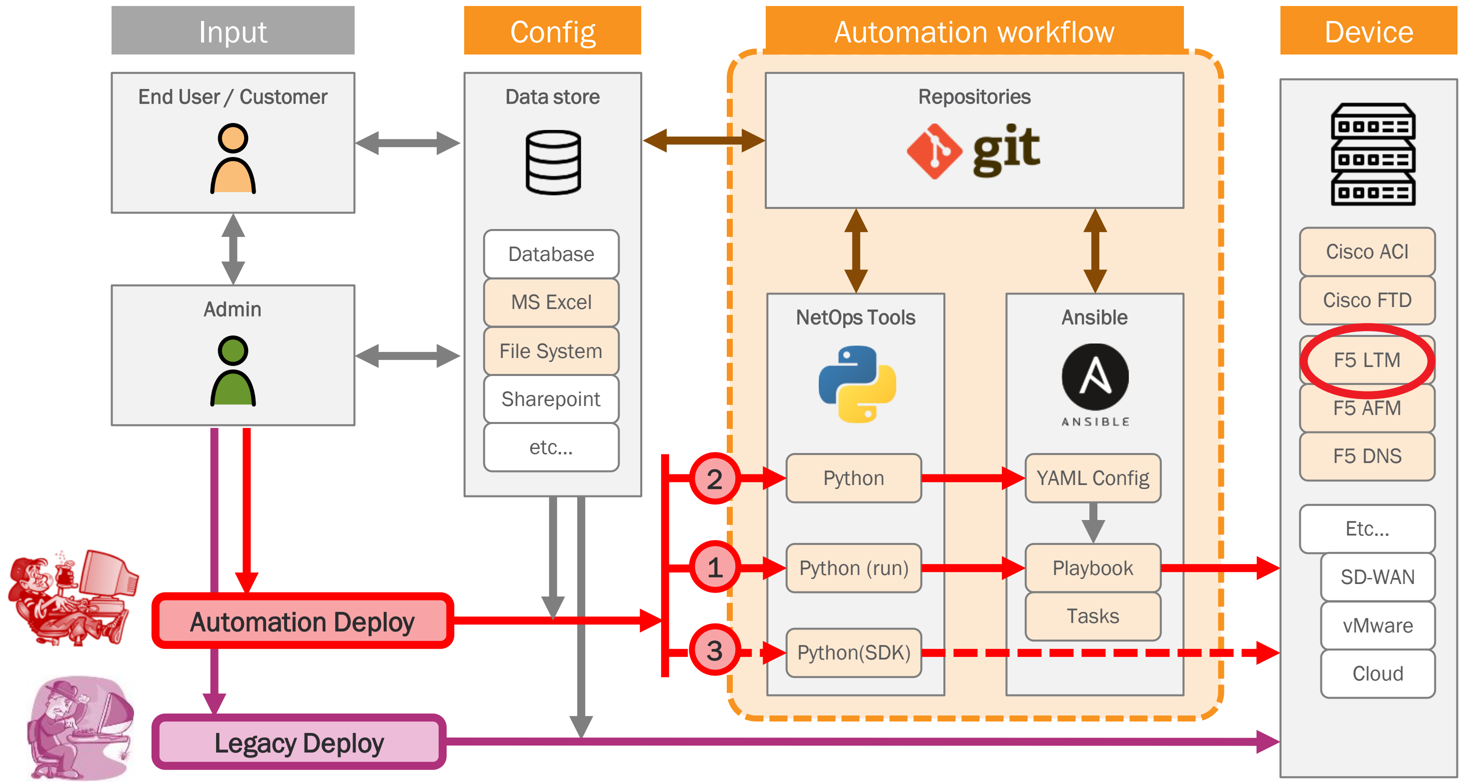
ANSIBLE python™ GitLab

Configuration/Automation workflow

SOITRONAUT
Soitron Automation Tool

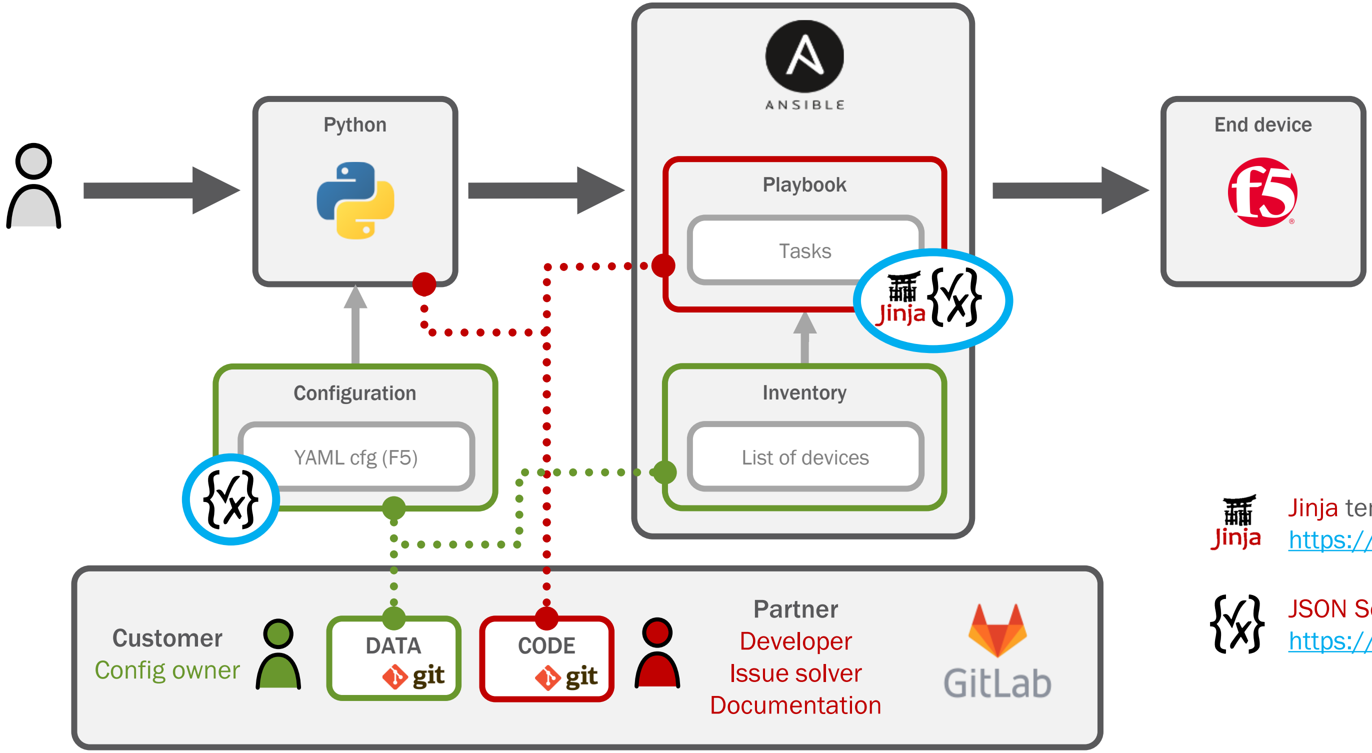


Configuration/Automation workflow



Ansible and Git (GitLab)

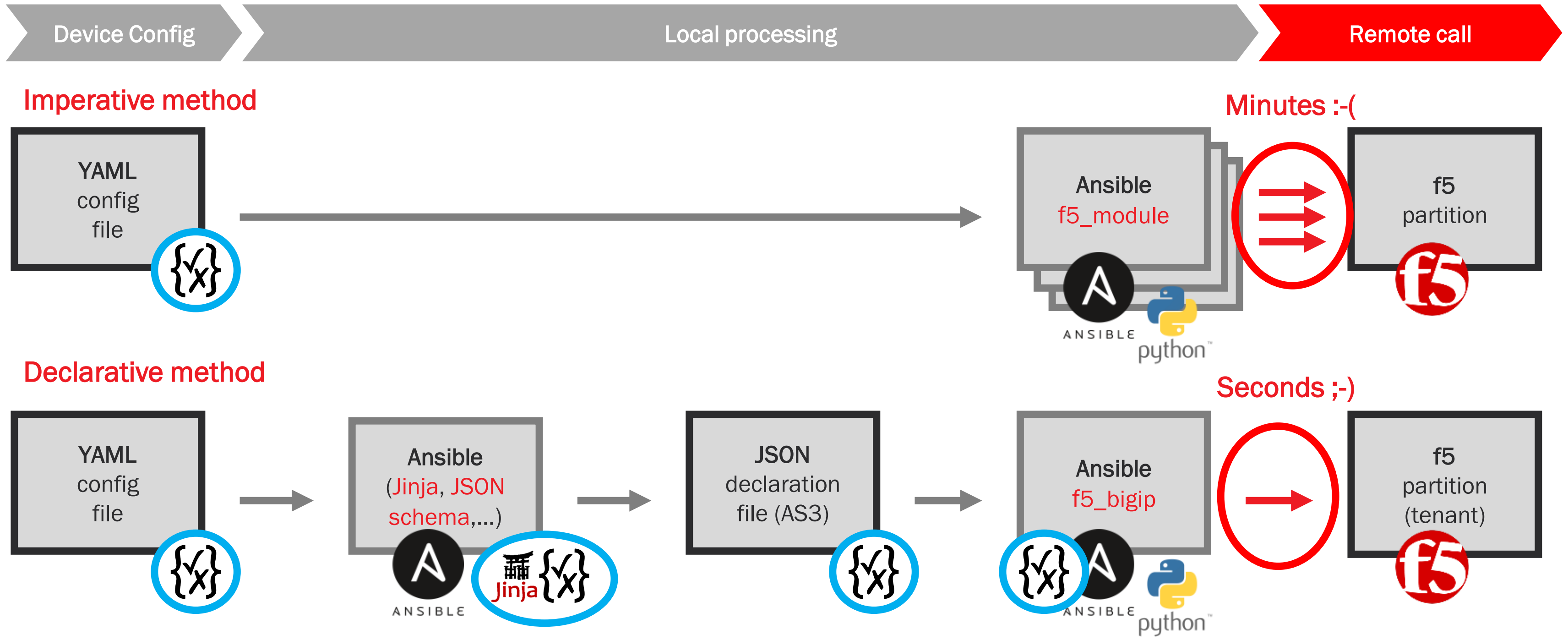
SOITRONAUT
Soitron Automation Tool



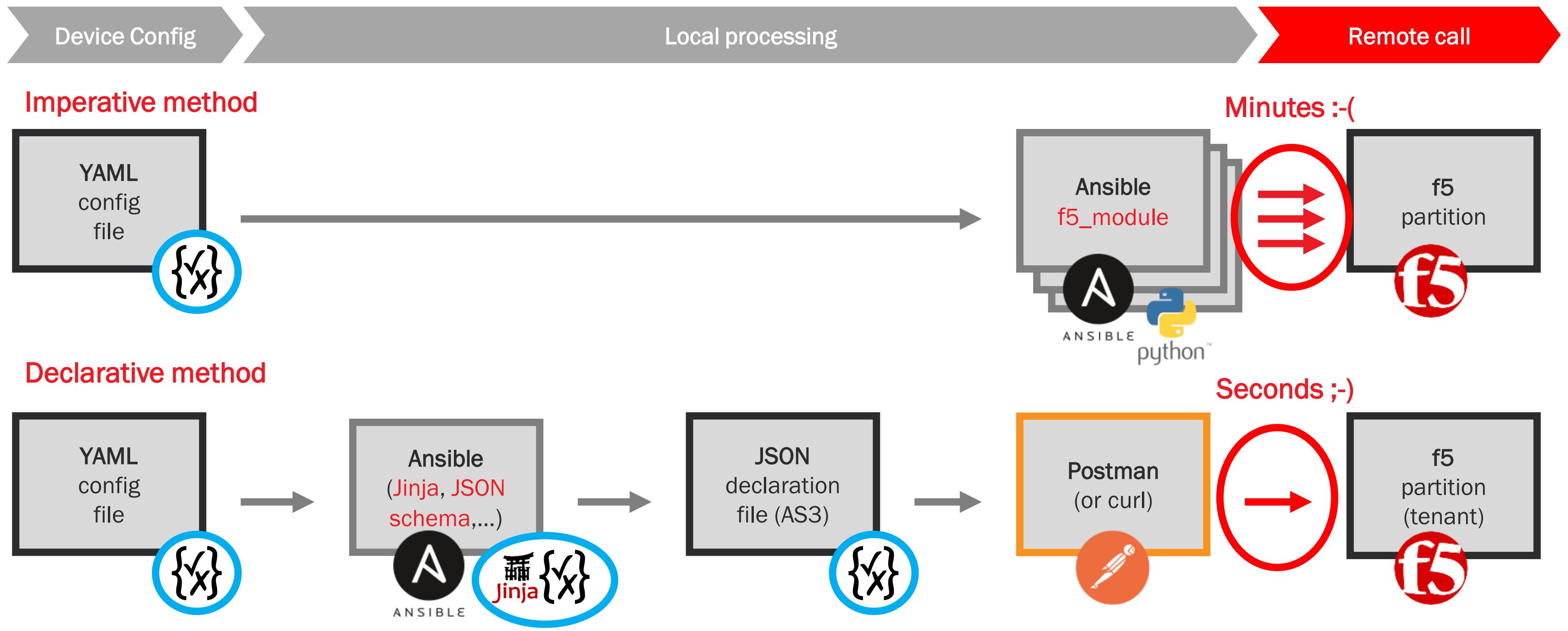
Jinja templating engine
https://jinja.palletsprojects.com/

JSON Schema declaration
https://json-schema.org/

Imperative vs. declarative model



Imperative vs. declarative model



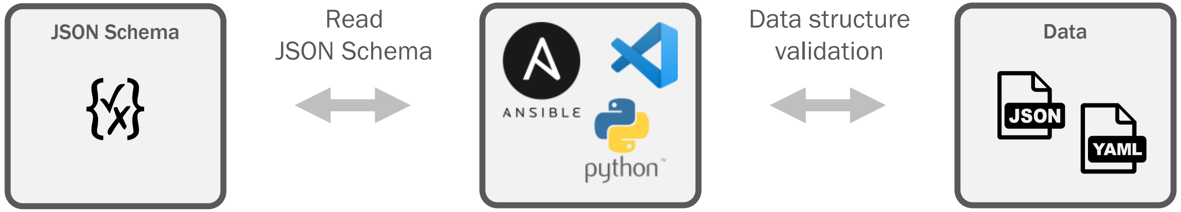
JSON Schema declaration

Benefits

- Describes your existing data format(s)
- Clear for human- and machine-readable documents
- Validates data which is useful for:
  - Automated testing
  - Ensuring quality of client submitted data

https://json-schema.org/

JSON Schema is a declarative language that allows you to annotate and validate JSON documents



Jinja templating engine

Jinja is a fast, expressive, extensible templating engine. Template is passed data to render the final document.

Jinja

https://jinja.palletsprojects.com/

YAML config:

```
hostname: "BIG-IP"
```

Ansible playbook:

```
---
- name: Write hostname
  hosts: all
  tasks:
  - name: write hostname using jinja2
    ansible.builtin.template:
      src: templates/test.j2
      dest: /tmp/output.txt
```

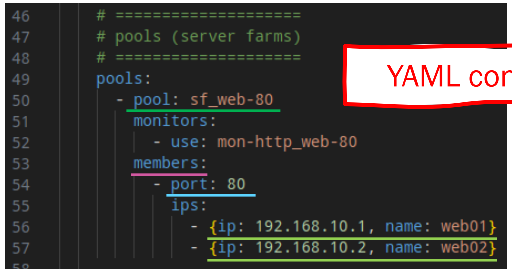
Jinja template (test.j2):

```
My name is {{ hostname }}!
```

Output file (/tmp/output.txt):

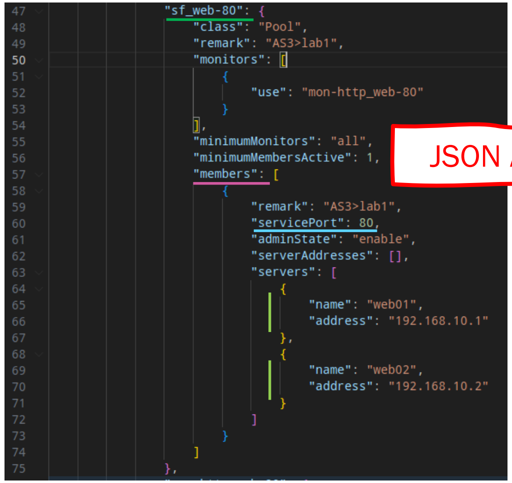
```
My name is BIG-IP!
```

Jinja F5 AS3 example



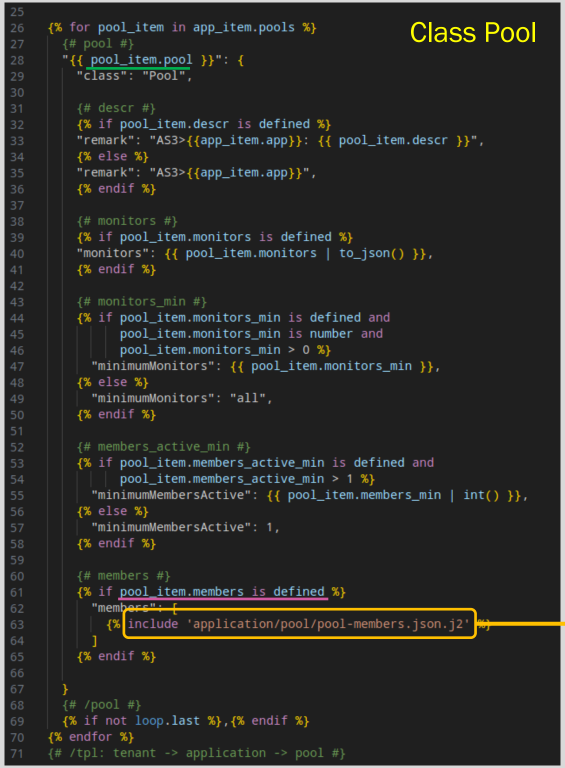
YAML config

```yaml
# ====================
# pools (server farms)
# ====================
pools:
  - pool: sf_web-80
    monitors:
      - use: mon-http_web-80
    members:
      - port: 80
        ips:
          - {ip: 192.168.10.1, name: web01}
          - {ip: 192.168.10.2, name: web02}
```



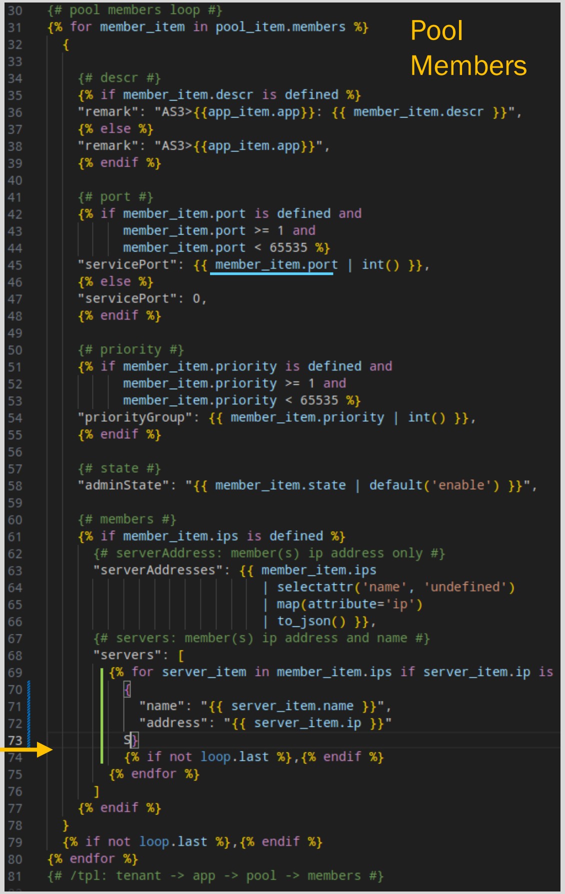
JSON AS3

```json
          "sf_web-80": {
            "class": "Pool",
            "remark": "AS3>lab1",
            "monitors": [
              {
                "use": "mon-http_web-80"
              }
            ],
            "minimumMonitors": "all",
            "minimumMembersActive": 1,
            "members": [
              {
                "remark": "AS3>lab1",
                "servicePort": 80,
                "adminState": "enable",
                "serverAddresses": [],
                "servers": [
                  {
                    "name": "web01",
                    "address": "192.168.10.1"
                  },
                  {
                    "name": "web02",
                    "address": "192.168.10.2"
                  }
                ]
              }
            ]
          },
```



Class Pool

```
{% for pool_item in app_item.pools %}
  {# pool #}
  "{{ pool_item.pool }}": {
    "class": "Pool",

    {# descr #}
    {% if pool_item.descr is defined %}
    "remark": "AS3>{{app_item.app}}: {{ pool_item.descr }}",
    {% else %}
    "remark": "AS3>{{app_item.app}}",
    {% endif %}

    {# monitors #}
    {% if pool_item.monitors is defined %}
    "monitors": {{ pool_item.monitors | to_json() }},
    {% endif %}

    {# monitors_min #}
    {% if pool_item.monitors_min is defined and
          pool_item.monitors_min is number and
          pool_item.monitors_min > 0 %}
      "minimumMonitors": {{ pool_item.monitors_min }},
    {% else %}
      "minimumMonitors": "all",
    {% endif %}

    {# members_active_min #}
    {% if pool_item.members_active_min is defined and
          pool_item.members_active_min > 1 %}
      "minimumMembersActive": {{ pool_item.members_min | int() }},
    {% else %}
      "minimumMembersActive": 1,
    {% endif %}

    {# members #}
    {% if pool_item.members is defined %}
      "members": [
        {% include 'application/pool/pool-members.json.j2' %}
      ]
    {% endif %}

  }
  {# /pool #}
  {% if not loop.last %},{% endif %}
{% endfor %}
{# /tpl: tenant -> application -> pool #}
```



Pool Members

```
{# pool members loop #}
{% for member_item in pool_item.members %}
  {

    {# descr #}
    {% if member_item.descr is defined %}
    "remark": "AS3>{{app_item.app}}: {{ member_item.descr }}",
    {% else %}
    "remark": "AS3>{{app_item.app}}",
    {% endif %}

    {# port #}
    {% if member_item.port is defined and
          member_item.port >= 1 and
          member_item.port < 65535 %}
    "servicePort": {{ member_item.port | int() }},
    {% else %}
    "servicePort": 0,
    {% endif %}

    {# priority #}
    {% if member_item.priority is defined and
          member_item.priority >= 1 and
          member_item.priority < 65535 %}
    "priorityGroup": {{ member_item.priority | int() }},
    {% endif %}

    {# state #}
    "adminState": "{{ member_item.state | default('enable') }}",

    {# members #}
    {% if member_item.ips is defined %}
      {# serverAddress: member(s) ip address only #}
      "serverAddresses": {{ member_item.ips
                            | selectattr('name', 'undefined')
                            | map(attribute='ip')
                            | to_json() }},
      {# servers: member(s) ip address and name #}
      "servers": [
        {% for server_item in member_item.ips if server_item.ip is
          {
            "name": "{{ server_item.name }}",
            "address": "{{ server_item.ip }}"
          S}
          {% if not loop.last %},{% endif %}
        {% endfor %}
      ]
    {% endif %}
  }
  {% if not loop.last %},{% endif %}
{% endfor %}
{# /tpl: tenant -> app -> pool -> members #}
```

Join at

slido.com

#soitron

Passcode: **tr4kxi**

SOITRONAUT
Soitron Automation Tool

Lab Time

- Lab #1: Simple service
  - Create
  - Modify
  - Delete
- Lab #2: Complex service
  - Create, Modify, Delete
  - Time comparison (Imperative vs Declarative)
- Lab #3: JSON Schema (+Jinja)
  - How to create and use it

LAB #1 – simple web service

Create (imperative/declarative)

- Imperative:
  - Step #1: Get active device
  - Step #2: Create Partition, Monitor, Pool, Profile, VS,… (many times)
- Declarative:
  - Step #1: Get active device
  - Step #2: Render JSON (AS3) (1x)
  - Step #3: Deploy (1x)

Modify (imperative/declarative)

- Imperative:
  - The same steps with delete combination (non-atomic)!
- Declarative:
  - Benefit: The same steps, but ATOMIC

Delete (declarative only)

- Imperative:
  - Delete each object (think about dependences)
- Declarative:
  - Delete all in one step

SOITRONAUT
Soitron Automation Tool

```
# how to use it
python3 runme.py –h

# apply imperative config
python3 runme.py --dev f5 ../data-example/lab1/

# apply declarative config
python3 runme.py --dev f5as3 ../data-example/lab1/

# delete partition
python3 runme.py --dev f5as3 ../data-example/lab1/ -t delete
```

LAB #2 – complex service

Create (imperative/declarative)

- Imperative:
  - Step #1: Get active device
  - Step #2: Create Partition, Monitor, Pool, Profile, VS,… (many times)
- Declarative:
  - Step #1: Get active device
  - Step #2: Render JSON (AS3) (1x)
  - Step #3: Deploy (1x)

Compare creation time

SOITRONAUT
Soitron Automation Tool

Modify (declarative)

- Imperative:
  - The same steps with delete combination (non-atomic)!
- Declarative:
  - The same steps, but ATOMIC

Delete (declarative)

- Imperative:
  - Delete each object (think about dependences)
- Declarative:
  - Delete all in one step

```
# how to use it
python3 runme.py -h

# apply imperative config
python3 runme.py --dev f5 ../data-example/lab2/

# apply declarative config
python3 runme.py --dev f5as3 ../data-example/lab2/

# delete partition
python3 runme.py --dev f5as3 ../data-example/lab2/ -t delete
```

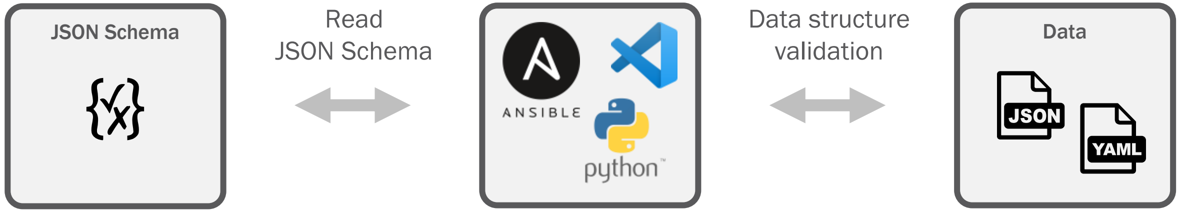
LAB #3 – JSON schema + Jinja

Simple JSON schema

- Validate JSON/YAML file using vscode
- How to create JSON schema file

Jinja example

- How to work with Jinja templating engine



SOITRONAUT

Soitron Automation Tool

Benefits

- ✓ The same workflow regardless end device
- ✓ Reducing human error
- ✓ Configuration without expert level knowledge
- ✓ Reducing configuration and/or migration time
- ✓ All configurations in GIT
- ✓ “Dry-run” – test before apply

F5 Advantages

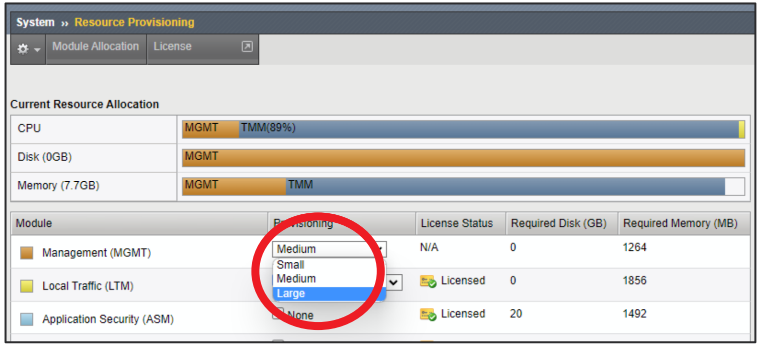
- ✓ Declarative model (AS3 JSON generator)
- ✓ Create/Change/Delete operation in seconds
- ✓ Ready for the future, ready for the BIG-IP Next

AS3

How to start with AS3
(some tips for beginners)

- Increase memory for restjavad
- AS3 software lifecycle (LTS, Feature…)
- Sync vs Async mode
- Dry-Run
- DELETE method

How to start with AS3 – read documentation


System » Resource Provisioning

Module Allocation | License

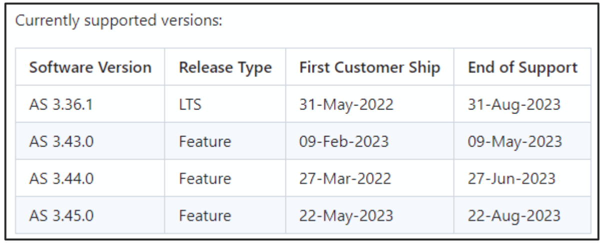
Current Resource Allocation

CPU	MGMT TMM(89%)
Disk (0GB)	MGMT
Memory (7.7GB)	MGMT TMM

Module	Provisioning	License Status	Required Disk (GB)	Required Memory (MB)
Management (MGMT)	Medium (Small / Medium / [✓ Large])	N/A	0	1264
Local Traffic (LTM)		Licensed	0	1856
Application Security (ASM)	None	Licensed	20	1492

Increase the restjavad memory allocation

- GUI: System > Resource Provisioning
  - On 14.x with 2+ modules select "Large"
  - Starting 15.x select "Large" always
- CLI: tmsh list sys db provision.extramb all-properties
- https://my.f5.com/manage/s/article/K26427018


Currently supported versions:

Software Version	Release Type	First Customer Ship	End of Support
AS 3.36.1	LTS	31-May-2022	31-Aug-2023
AS 3.43.0	Feature	09-Feb-2023	09-May-2023
AS 3.44.0	Feature	27-Mar-2022	27-Jun-2023
AS 3.45.0	Feature	22-May-2023	22-Aug-2023

Install f5-appsvc-extension

- https://github.com/F5Networks/f5-appsvcs-extension
- Look to Postman collection with examples
- Read Release notes :-)

Check AS3 Software Lifecycle

- https://github.com/F5Networks/f5-appsvcs-extension/blob/master/SUPPORT.md

RTFM! ;-)
https://clouddocs.f5.com/products/extensions/f5-appsvcs-extension/latest/userguide/

Read FAQ
https://clouddocs.f5.com/products/extensions/f5-appsvcs-extension/latest/userguide/faq.html

How to start with AS3 – good to know

Sync vs Asynchronous mode

- Introduced in AS3 3.7.0
- Even if async mode is set to false, after 45 seconds BIG-IP AS3 sets asynchronous mode to true (API swap, HTTP 202)
- https://clouddocs.f5.com/products/extensions/f5-appsvcs-extension/latest/refguide/as3-api.html#api-methods
- Example:
  - POST https://192.0.2.10/mgmt/shared/appsvcs/declare
  - POST https://192.0.2.10/mgmt/shared/appsvcs/declare?async=true

DELETE method

- Remove configurations for one, more or all declared Tenants from the target ADC
- Example:
  - DELETE https://192.0.2.10/mgmt/shared/appsvcs/declare/T1
  - DELETE https://192.0.2.10/mgmt/shared/appsvcs/declare/T1,T2,T5
  - DELETE https://192.0.2.10/mgmt/shared/appsvcs/declare

Dry run

- Similar to the deploy action, dry-run sends the declaration through all validation checks but does not attempt to deploy the configuration on the device
- Example:
  - POST https://192.0.2.10/mgmt/shared/appsvcs/declare?controls.dryRun=true

Be ready for the future

BIG-IP

- HW product lifecycle (EoL is coming: ~2025-2031)
  https://my.f5.com/manage/s/article/K4309
- SW development and support up to 17.5.x.x
  https://my.f5.com/manage/s/article/K5903
- Limited SW support for new platforms (rSeries, VELOS)

Are you ready for the future?

BIG-IP Next

- New platforms: rSeries, VELOS
- New SW releases naming schema
  https://my.f5.com/manage/s/article/K000135785
- Management: no GUI -> BIG-IP Next Central Manager (CM)
- Modules: LTM and WAF (additional coming soon)
- Automation tools (AS3, DO, TS, FAST): no longer requires download/install
- Ability to reuse:
  - existing BIG-IP AS3 declarations with BIG-IP Next
  - existing automation tools!

SOITRONAUT
Soitron Automation Tool

Questions?

Join at

slido.com

#soitron

Passcode: **tr4kxi**

Praha

Soitron s.r.o.

Pekařská 621/7

155 00 Praha 5

tel.: +420 266199918

Bratislava

Soitron, s.r.o.

Plynárenská 5

829 75 Bratislava 25

tel.: +421 258224111

e-mail: marketing@soitron.com
web: www.soitron.com

Martin Kyrc
Senior Network System Engineer
martin.kyrc@soitron.com

SOITRON*