# Nornir: Solve Big Problems Fast

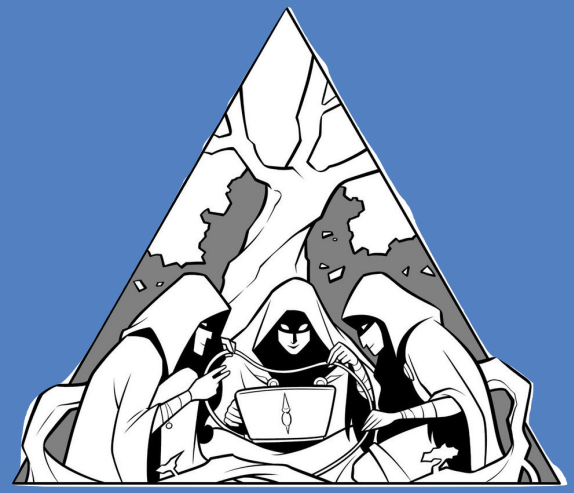
Brett Lykins
Managing Consultant @ Network To Code
Twitter: @lykinsb - GitHub: lykinsbd

# Agenda

- What is Nornir?
- Nornir vs. ???
- Nornir Concepts
- Use Cases

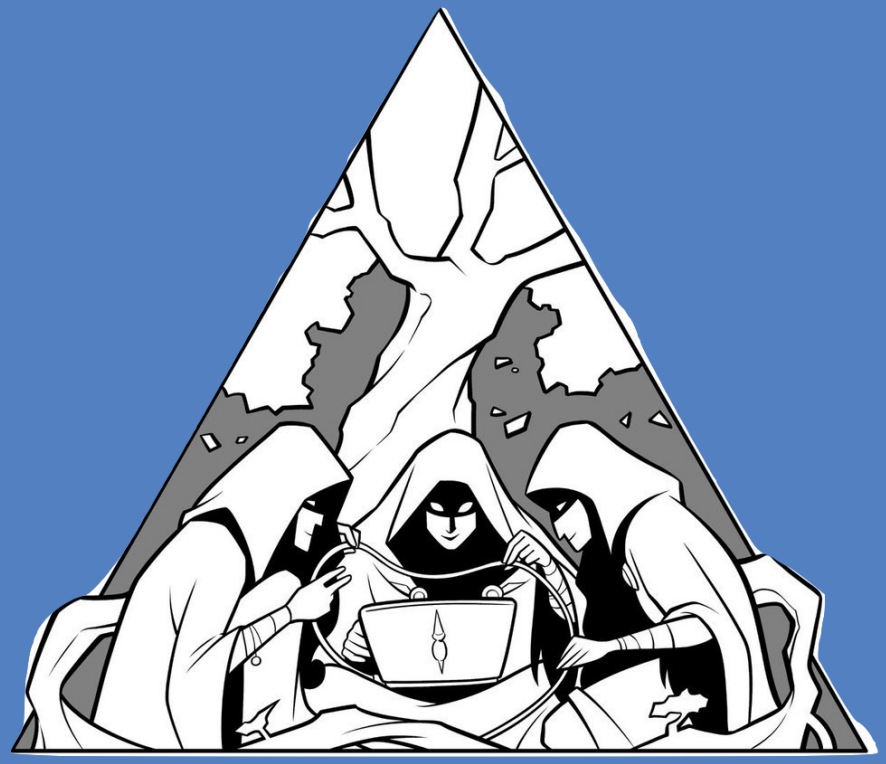
# What is Nornir?

- Python automation framework
- Inventory of hosts
- Tasks executed against those hosts
- Highly extensible/pluggable
- Agentless
- https://github.com/nornir-automation/nornir/



# Basic Stats

- Started in December 2017
- 2.0.0 in December 2018
- Current release: 2.5.0
- 35 contributors
- 644 stars on Github



# What About Ansible?

Isn’t Ansible a Python based automation tool?

# Nornir Ain't Ansible

- No Domain Specific Language (DSL)
- Logic kept in code, not YAML files
- **100% PYTHON**
  - Use existing Python linters/IDEs
  - Access to native Python debugging

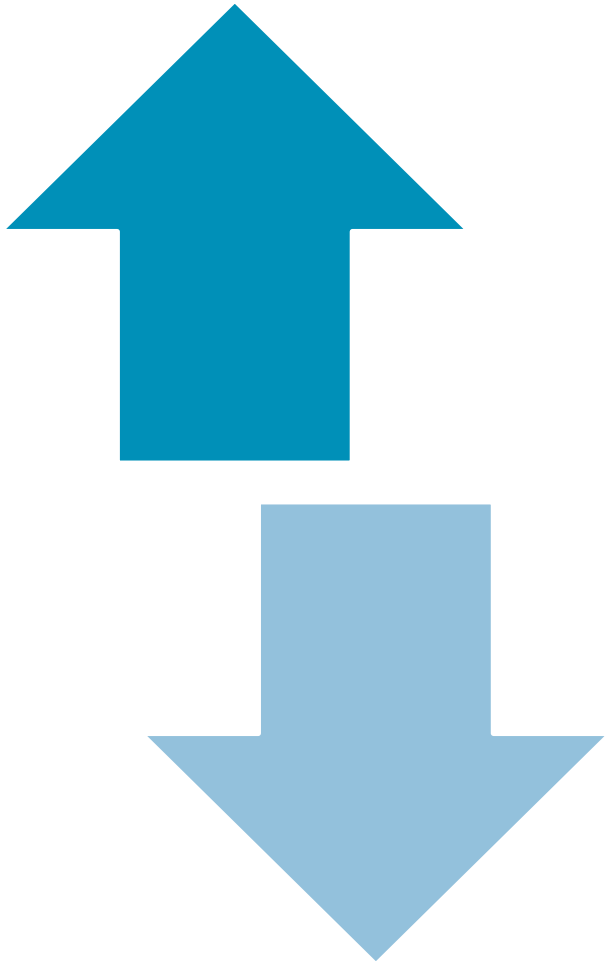
# Ansible Has A Place

- Lower barrier to entry
  - Community of available modules and playbooks
  - Widely used for enterprise server management
  - May already be in use in your organization
- AWX/Tower for orchestration/execution
- “Batteries Included”

# “What about this other tool…”

- Nornir focused initially on Network devices
- Founders from Network Automation community
  - Maintainers of Netmiko and NAPALM
- Sought to fill gaps in existing tooling
  - Ansible had historically lackluster network device support (gotten better in recent years)
  - Salt (and others) required agent or proxy hosts

# Pros and Cons



- Pros
  - Native Python tooling
  - Easily extensible
  - First-class support for network devices
- Cons
  - Small (but growing) community
  - Less “Batteries Included”

# Nornir Key Benefits

- Easily integrated with other Python frameworks
- Expose tasks via Flask based API
- Integrate with an existing Django based web app
  - Allows simple NetBox plugin creation
  - https://github.com/netbox-community/netbox

# Nornir Key Benefits

- While speed isn't everything… Nornir is FAST!
- Several orders of magnitude faster than Ansible
- Shoutout to Patrick Ogenstad
  - https://networklore.com/ansible-nornir-speed

# Nornir Concepts

# The Good Stuff

- Inventory
  - Hosts
  - Groups
  - Defaults

# The Good Stuff

- Tasks
  - Actions on hosts
  - Essentially Python functions
    - But with some free goodies/magic

# Inventory Plugins

- Simple Inventory (YAML)
- Ansible
- Network Source of Truth (NSOT)
- NetBox

# Inventory Plugins

- Write your own!
- Plugin API is easy to use
- https://nornir.readthedocs.io/en/latest/howto/writing_a_custom_inventory_plugin.html

# Simple Inventory Plugin

- Most common Inventory type seen
- YAML based
- Separate hosts.yml and groups.yml files

# Simple Inventory Plugin

- Hosts file example:

```
$cat inventory/hosts.yml
---
asav1:
  hostname: 10.10.10.50
  password: cisco
  groups:
    - asavs
  data:
    arbitrary_data_1 = “A”
```

# Simple Inventory Plugin

- Groups file example:

```
$cat inventory/groups.yml
---
asavs:
  platform: cisco_asa
  username: cisco
  data:
    arbitrary_data_2: “B”
```

# Nornir Tasks

- Python functions executed against each host
- Tasks must accept at least a *nornir.core.task.Task* object
- Example:

```
def mah_task(task):
    print(
        f“Doin’ stuff to {task.host.name}”
    )
```

# Nornir Tasks

- Can be grouped or nested inside each other
- Create your own
- Or use some of the common tasks included with Nornir
  - https://nornir.readthedocs.io/en/latest/plugins/tasks/index.html
  - Local file operations
  - Text manipulation
  - Network device or server command execution

# Nornir Tasks

- Builtin tasks are in *nornir.plugins.tasks*
- Send a command to a server via Paramiko SSH connection:
  - *commands.remote_command*
- Send a command or configuration to a network device via Netmiko:
  - *networking.netmiko_send_command*
  - *networking.netmiko_send_config*
- Gather network device configuration via NAPALM:
  - *networking.napalm_get*

# Let's Get Started: Initialization

- Initialization
  - *nornir.InitNornir()*
  - *Nornir* object instantiation
  - Read inventory and configuration
  - Prepare for task execution
  - Filtering

# Let's Get Started: Initialization

- Flexible configuration
  - Config file (more YAML!)
  - Parameters to *nornir.InitNornir()*
- Config options
  - How many worker threads
  - Where are we logging to
  - Which inventory style/plugin
    - Where can I find the inventory

# Configuration File Example

```
---
core:
  num_workers: 100
logging:
  file: “/Users/brett.lykins/nornir/logs/nornir.log"
  loggers: ["nornir", "paramiko", "netmiko"]
inventory:
  plugin: nornir.plugins.inventory.simple.SimpleInventory
  options:
    host_file: “/Users/brett.lykins/nornir/inventory/hosts.yml"
    group_file: “/Users/brett.lykins/nornir/inventory/groups.yml"
ssh:
  config_file: “/Users/brett.lykins/nornir/ssh_config"
```

# Let's Get Started: Initialization

- Config File:

```
from nornir import InitNornir
mah_nornir = InitNornir(
    config_file="config.yaml”
)
```

# Let's Get Started: Initialization

- Config File and Parameters:

```
from nornir import InitNornir
mah_nornir = InitNornir(
    core={"num_workers": 50},
    config_file="config.yaml"
)
```

# Filtering Your Inventory

- Filter on any metadata in the inventory
- Pass kwargs to *Nornir.filter():*

```
filtered_hosts = mah_nornir.filter(
    vendor="Cisco", is_colo=False
)
```

# Filtering Your Inventory

- Call a function via *Nornir.filter(filter_func=mah_filter)*

```
def mah_filter(host):
    return host.data[”device_id"] in args.firewall_ids

filtered_hosts = mah_nornir.filter(
    filter_func=mah_filter
)
```

- See docs for more examples and details on *F* filtering

# And Away We Go

- Execute tasks on hosts
  - *Nornir.run(mah_task)*
  - Inventory (or filtered subset)
  - Returns a *Results* dict-like object

# Example Execution

```
>>> from nornir import InitNornir
>>> mah_nornir =
InitNornir(config_file="config.yaml")
>>> def mah_task(task):
...     print(f"Doin' stuff to
{task.host.name}!")
>>> mah_nornir.run(mah_task)
Doin' stuff to labhost.test!
```

# Review

- Inventory of hosts
  - Host/Group/Default attributes and data
- Tasks to execute on those hosts
  - Many pre-canned
- Create a configuration file
  - Or just pass attributes to *InitNornir()*
- Initialize a *Nornir* instance
- *mah_nornir.run(mah_task)*

# Use Cases

# Company X Buys Company Y

- Company X has robust Network Automation Platform
  - Tens of Thousands of Devices
  - Python/Flask based APIs and web services
  - Reporting/Changes to Devices
- Company Y… Not so much.

# Company X Buys Company Y

- Existing tooling not compatible with Company Y
  - Different CMDB/Inventory systems
  - Different authentication systems
- Auditors and Company X leadership expect similar experience/results from Company Y network
- Extremely tight deadlines and no $$$

# Nornir To The Rescue

- Extract Company Y inventory into Nornir Simple Inventory YAML format
- Utilize existing Python applications and Flask APIs
- Build Views/Routes specifically for Company Y
- Completed in under 1 week

# Company D: “We’ve got NetBox, now what?”

- NetBox installed as DCIM tool
  - Populated with all network devices
- Desire to leverage NetBox for automation
  - 6+ week lead time on additional VMs or Load Balancer/Firewall config
  - Unfamiliar with Python web frameworks

# Nornir = Good To Go

- Use existing NetBox VM and associated infrastructure
- Deploy with existing Django app as an additional view
  - Use *django-rq* for job processing
- Lower barrier to entry
- Simplify deployment
- POC to Production in under one month

# How Can I Get Nornir?

- Github: https://github.com/nornir-automation/nornir
- Pypi: pip install nornir

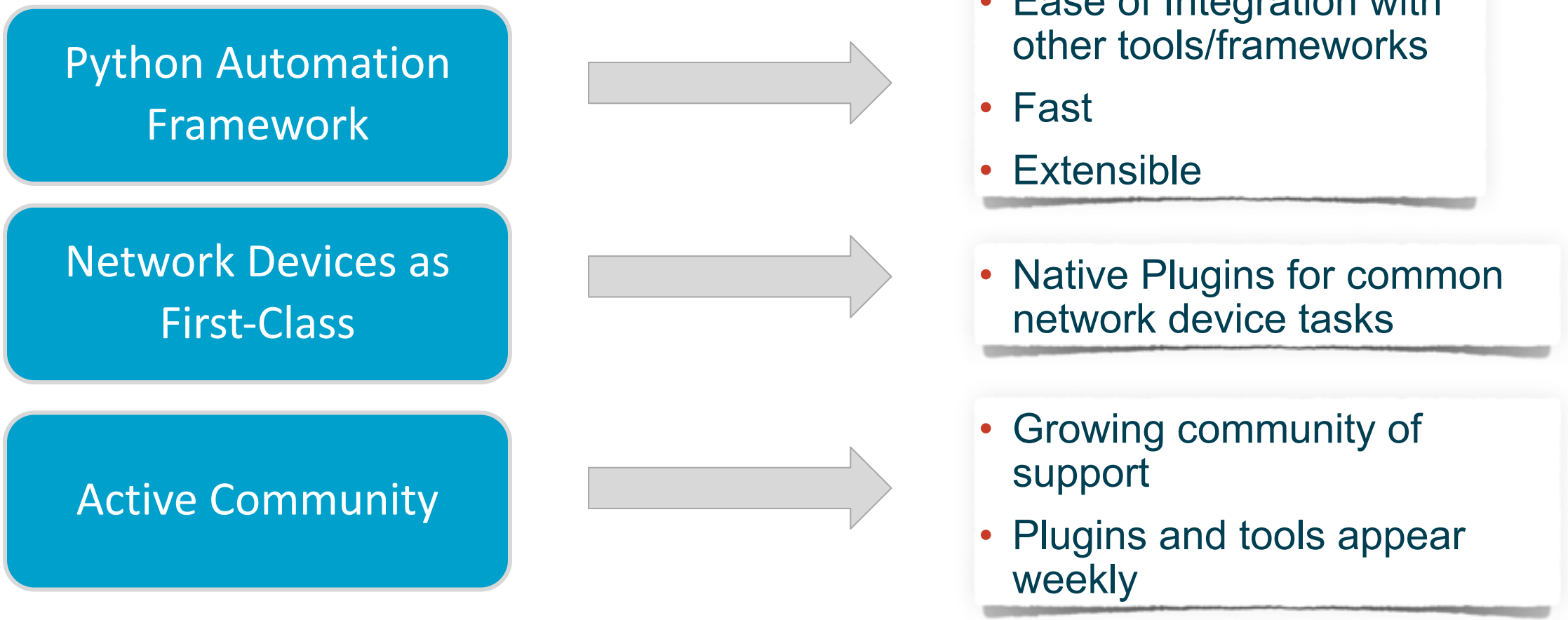
# Where can I learn more?

- Docs: https://nornir.readthedocs.io
- Slack: http://slack.networktocode.com/
  - #nornir
- Discourse: https://nornir.discourse.group/

# Nornir 3.0: What does the future hold?

- Under active development
  - https://github.com/nornir-automation/nornir/milestone/1
- Separation of plugins into independent repositories and packages
  - Listing of available community Plugins
    - https://nornir.tech/nornir/plugins/
- Only install the plugins you care about
  - *pip install nornir nornir-netmiko nornir-netbox*

# Nornir in Summary



# Thank You

Brett Lykins
Managing Consultant @ Network To Code
Twitter: @lykinsb
GitHub: lykinsbd