

Rollout all your Prometheus exporters with Puppet!

(Proudly powered by Vox Pupuli Puppet modules)

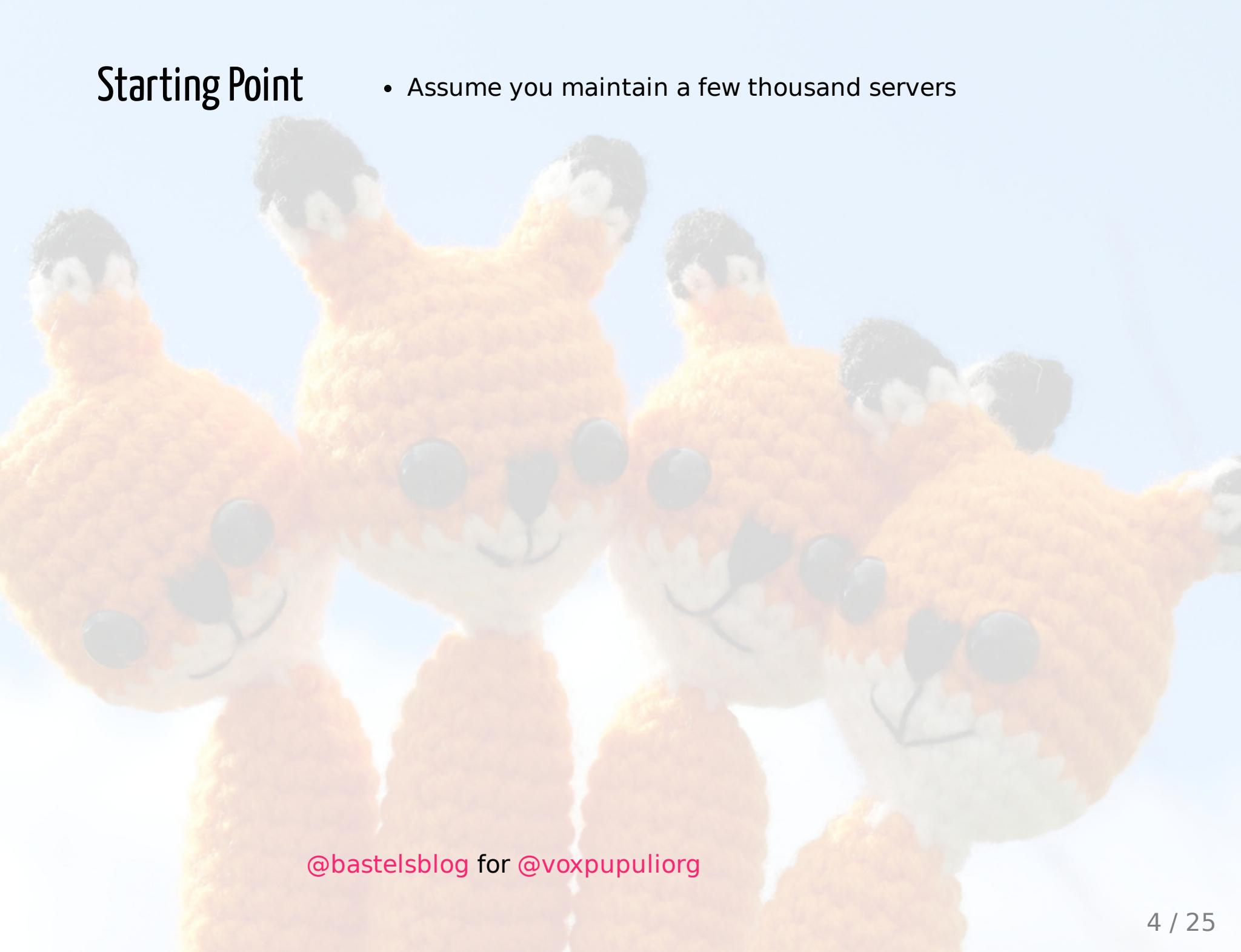
\$ whoami

- Tim 'bastelfreak' Meusel
- Puppet Contributor since 2012
- Merging stuff on [Vox Pupuli](#) (Puppet Community) since 2015
- Vox Pupuli Project Management Committee member

Let us go on a monitoring journey

Starting Point

- Assume you maintain a few thousand servers



@bastelsblog for @voxpupuliorg

Starting Point

- Assume you maintain a few thousand servers
- Assume you have at least one coworker that develops software

@bastelsblog for @voxpupuliorg

Starting Point

- Assume you maintain a few thousand servers
- Assume you have at least one coworker that develops software
- You have to maintain that software in production

@bastelsblog for @voxpupuliorg

Starting Point

- Assume you maintain a few thousand servers
- Assume you have at least one coworker that develops software
- You have to maintain that software in production
- All monitoring you have is really bad
 - It has false positives, it isn't efficient, it's complex

@bastelsblog for @voxpupuliorg

Starting Point

- Assume you maintain a few thousand servers
- Assume you have at least one coworker that develops software
- You have to maintain that software in production
- All monitoring you have is really bad
 - It has false positives, it isn't efficient, it's complex
- Or you have no monitoring at all

@bastelsblog for @voxpupuliorg

Chapter 1: Exploration

Or: The Happy Phase

Starting Point

- You attend your most favourite monitoring conference

Exploration

@bastelsblog for @voxpupuliorg

Starting Point

- You attend your most favourite monitoring conference
- Somebody recommends you `node_exporter`
- "Just download the binary, run it as root and it works" they said...

Exploration

@bastelsblog for @voxpupuliorg

Starting Point

Exploration

Prometheus Exporters:

- Reads many metrics from your box, expose them via HTTP
 - No HTTPS, no authentication

Starting Point

Exploration

Prometheus Exporters:

- Reads many metrics from your box, expose them via HTTP
 - No HTTPS, no authentication
- Has many collectors, that can be enabled/disabled

Starting Point

Exploration

Prometheus Exporters:

- Reads many metrics from your box, expose them via HTTP
 - No HTTPS, no authentication
- Has many collectors, that can be enabled/disabled
- Can read values from 3rd parties from a txt file

Starting Point

Exploration

Prometheus Exporters:

- Reads many metrics from your box, expose them via HTTP
 - No HTTPS, no authentication
- Has many collectors, that can be enabled/disabled
- Can read values from 3rd parties from a txt file
- Single binary, written in go
 - Nice for dependency management, but please consider building packages
 - Do not download random binaries from the internet...

Starting Point

Exploration

Prometheus Exporters:

```
# HELP Total number of scrapes by HTTP status code.  
# TYPE counter  
promhttp_metric_handler_requests_total{code="200"} 5  
promhttp_metric_handler_requests_total{code="500"} 0  
promhttp_metric_handler_requests_total{code="503"} 0
```

- Each metric has a name and a value
- There should be a HELP
- There should be a type
- A metric can have labels (code="XXX" in this case)

Starting Point

- Tell your developers to expose data as prometheus metrics

Exploration

@bastelsblog for @voxpupuliorg

Starting Point

- Tell your developers to expose data as prometheus metrics
- Enrich your exporter with 3rd party data

Exploration

@bastelsblog for @voxpupuliorg

Starting Point

- Tell your developers to expose data as prometheus metrics
- Enrich your exporter with 3rd party data
- Many applications already provide prometheus-style metrics

Exploration

@bastelsblog for @voxpupuliorg

Starting Point

- Tell your developers to expose data as prometheus metrics
- Enrich your exporter with 3rd party data
- Many applications already provide prometheus-style metrics
- There are **hundreds of exporters** available

Exploration

@bastelsblog for @voxpupuliorg

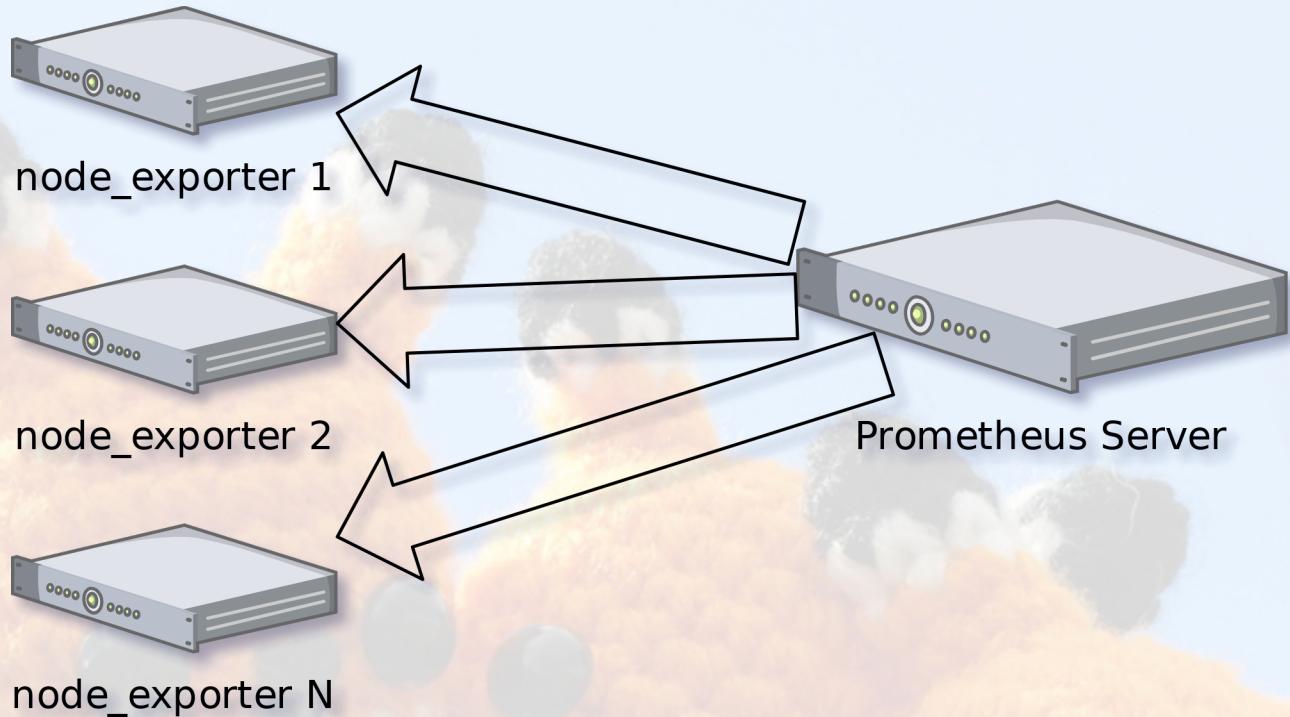
Chapter 2: Integration

Or: Million ways to shoot yourself in the foot

Starting Point

Exploration

Integration



With the **puppet/prometheus** module:

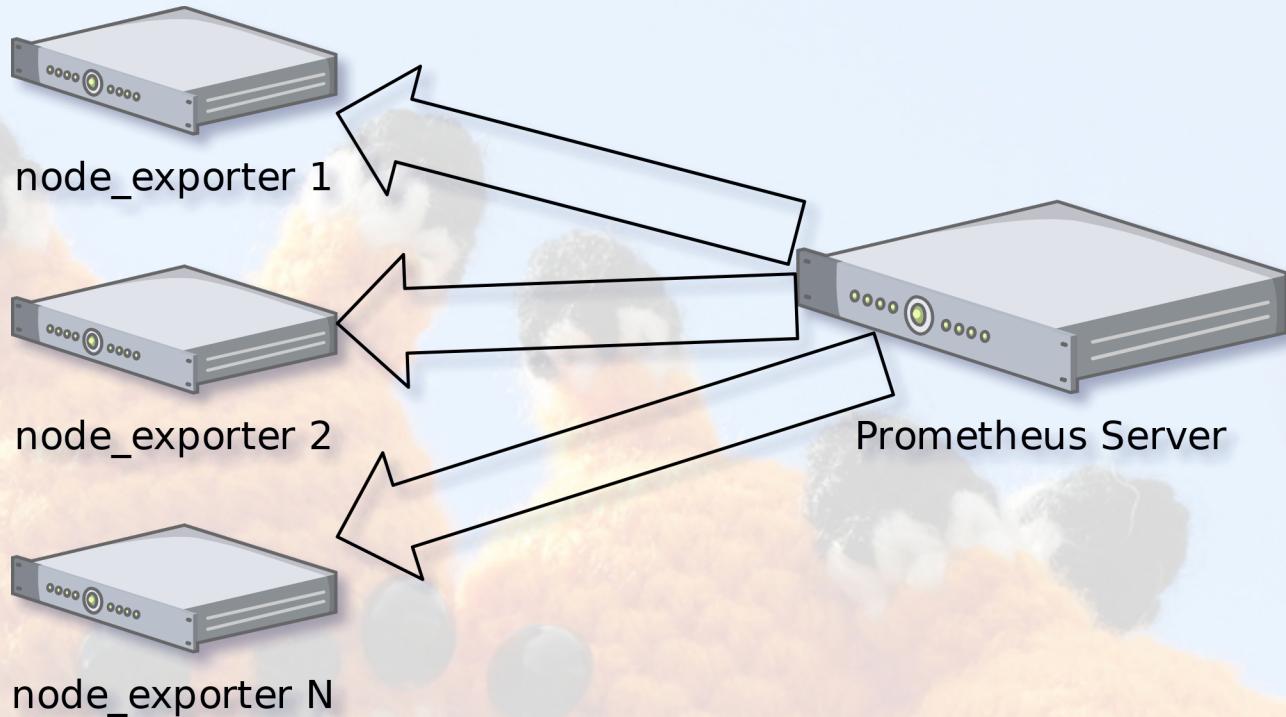
```
include prometheus::node_exporter
```

@bastelsblog for @voxpupuliorg

Starting Point

Exploration

Integration



- Prometheus database scrapes all exporters
- Everybody can make backups of your metrics \o/
 - (Because they are available to the whole internet)

Starting Point

- Prometheus database needs to know the targets
- Register each exporter in Consul as a service
- Feed services into Prometheus
- Use [KyleAnderson/consul](#) Puppet module to deploy consul

Exploration

Integration

Discovery

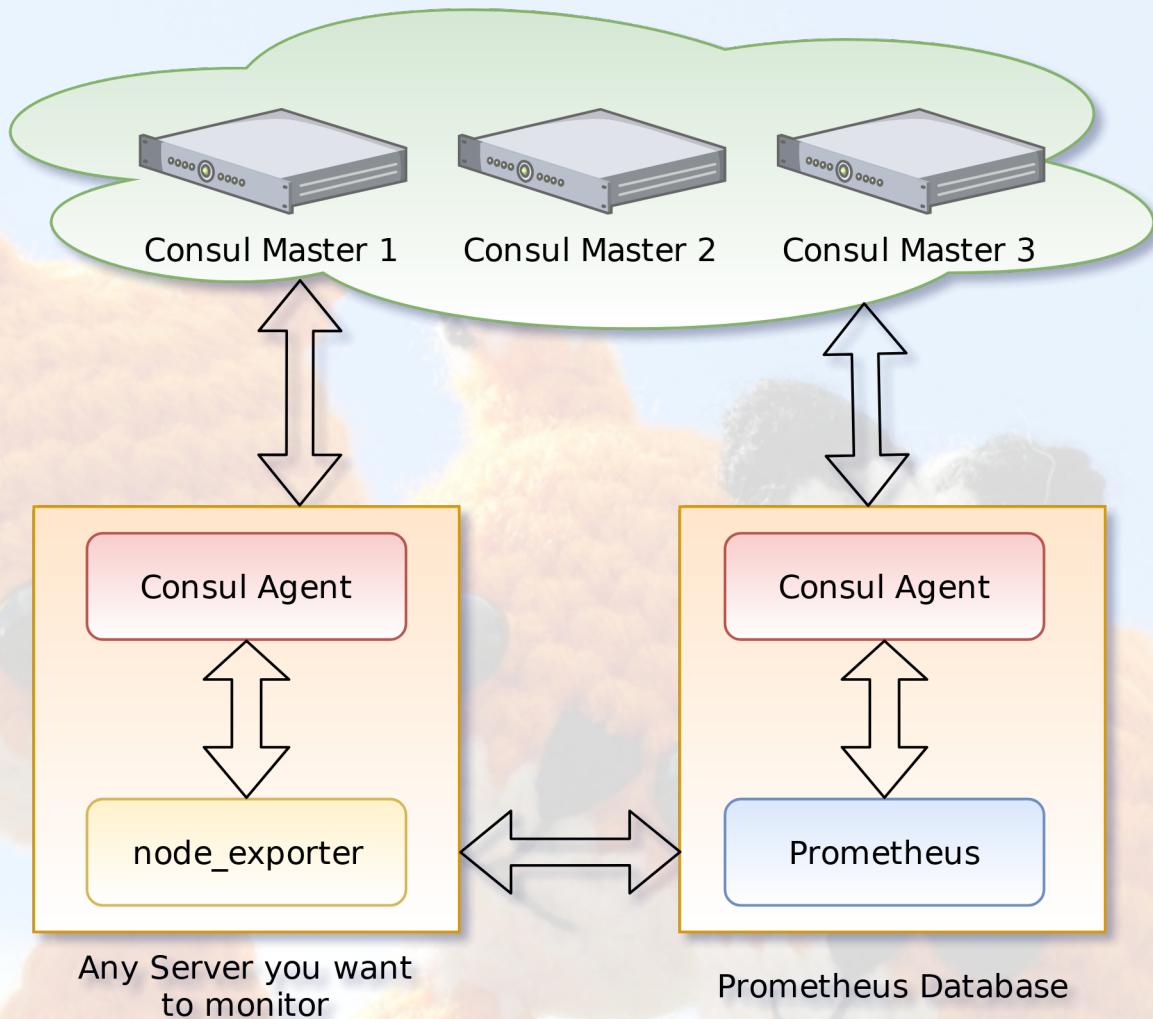
@bastelsblog for @voxpupuliorg

Starting Point

Exploration

Integration

Discovery



@bastelsblog for @voxpupuliorg

Starting Point

Take a look at [./examples/node_exporter_consul.pp](#)

Exploration

Integration

Discovery

@bastelsblog for @voxpupuliorg

Starting Point

- Puppet Agent/Server use TLS client certificates
- Deploy nginx in front of node_exporter
- Require a valid client certificate from prometheus database
- Vox Pupuli provides `puppet/nginx \o/`

Exploration

Integration

Discovery

Authentication

@bastelsblog for @voxpupuliorg

Starting Point

Take a look at `./examples/node_exporter_profile.pp`

Exploration

Integration

Discovery

Authentication

@bastelsblog for @voxpupuliorg

Starting Point

Exploration

Integration

Discovery

Authentication

Consul:

- No authentication by default
- Can also use TLS client certificates / a secret key
- Puppet certificates can be reused

Starting Point

- Only Prometheus database should be able to access exporters

Exploration

Integration

Discovery

Authentication

Firewalling

@bastelsblog for @voxpupuliorg

Starting Point

- Only Prometheus database should be able to access exporters
- Firewalling should be automated

Exploration

Integration

Discovery

Authentication

Firewalling

@bastelsblog for @voxpupuliorg

Starting Point

- Only Prometheus database should be able to access exporters

Exploration

- Firewalling should be automated
- IP addresses might change often

Integration

Discovery

Authentication

Firewalling

Starting Point

Exploration

Integration

Discovery

Authentication

Firewalling

```
class profiles::prometheus {
    @@erm::rule{"prometheus2$trusted['certname']":
        ensure => 'present',
        chain  => 'INPUT',
        saddr   => facts['networking']['ip6'],
        dport   => 9100,
        proto   => 'tcp',
        policy  => 'ACCEPT',
        tag     => 'prometheus2node_exporter',
    }
}

class profiles::node_exporter {
    Erm::Rule <<| title == 'prometheus2node_exporter' |>>
}
```

- Exported resources to the rescue
- Vox Pupuli provides puppet/erm

Starting Point

Exploration

Integration

Discovery

Authentication

Firewalling

Consul:

- Consul creates a meshed network
- iptables doesn't scale with thousands of entries
- filtering packets and changing rules will be delayed

Starting Point

ipsets and separate chains to the rescue!

Exploration

- handle consul traffic in a separate iptables chain

- ip sets are a hashmap of CIDR ranges

- In memory, scale very good

- iptables can check against this hashmap

- You cannot mix IPv4/IPv6 in one hashmap

- Updating the ip set is independent from iptables rules

- Vox Pupuli provides [puppet/ipset](#)

Firewalling

Starting Point

Take a look at [./examples/ipsets.pp](#)

Exploration

Integration

Discovery

Authentication

Firewalling

@bastelsblog for @voxpupuliorg

Chapter 3: Conclusion

Or: \o/ ?

Starting Point

- Rolling out an exporter with Puppet is easy
 - 1 line of code! (for a dev environment)
 - Required config management code for a production setup is still reasonable

Exploration

Integration

Conclusion

@bastelsblog for @voxpupuliorg

Starting Point

- Rolling out an exporter with Puppet is easy
- `puppet/prometheus` supports 20 famous exporters
 - You can easily add support for another one
 - Ping the `Vox Pupuli` people and ask for help

Exploration

Integration

Conclusion

@bastelsblog for @voxpupuliorg

Starting Point

- Rolling out an exporter with Puppet is easy
- `puppet/prometheus` supports 20 famous exporters
- Consul as service discovery
 - + Probably the solution that scales best
 - + Since you now have it, you can use it for even more stuff
 - More configuration effort than an actual exporter

Exploration

Integration

Conclusion

Starting Point

- Rolling out an exporter with Puppet is easy
- [puppet/prometheus](#) supports 20 famous exporters
- Consul as service discovery
- [github.com/bastelfreak/osmc2019](#) repo with slides + all examples + Vagrantfile to show your coworkers
- Contact: tim@bastelfreak.de or bastelfreak on freenode
- [Collection of related talks](#)

Thanks for your attention!

@bastelsblog for @voxpupuliorg