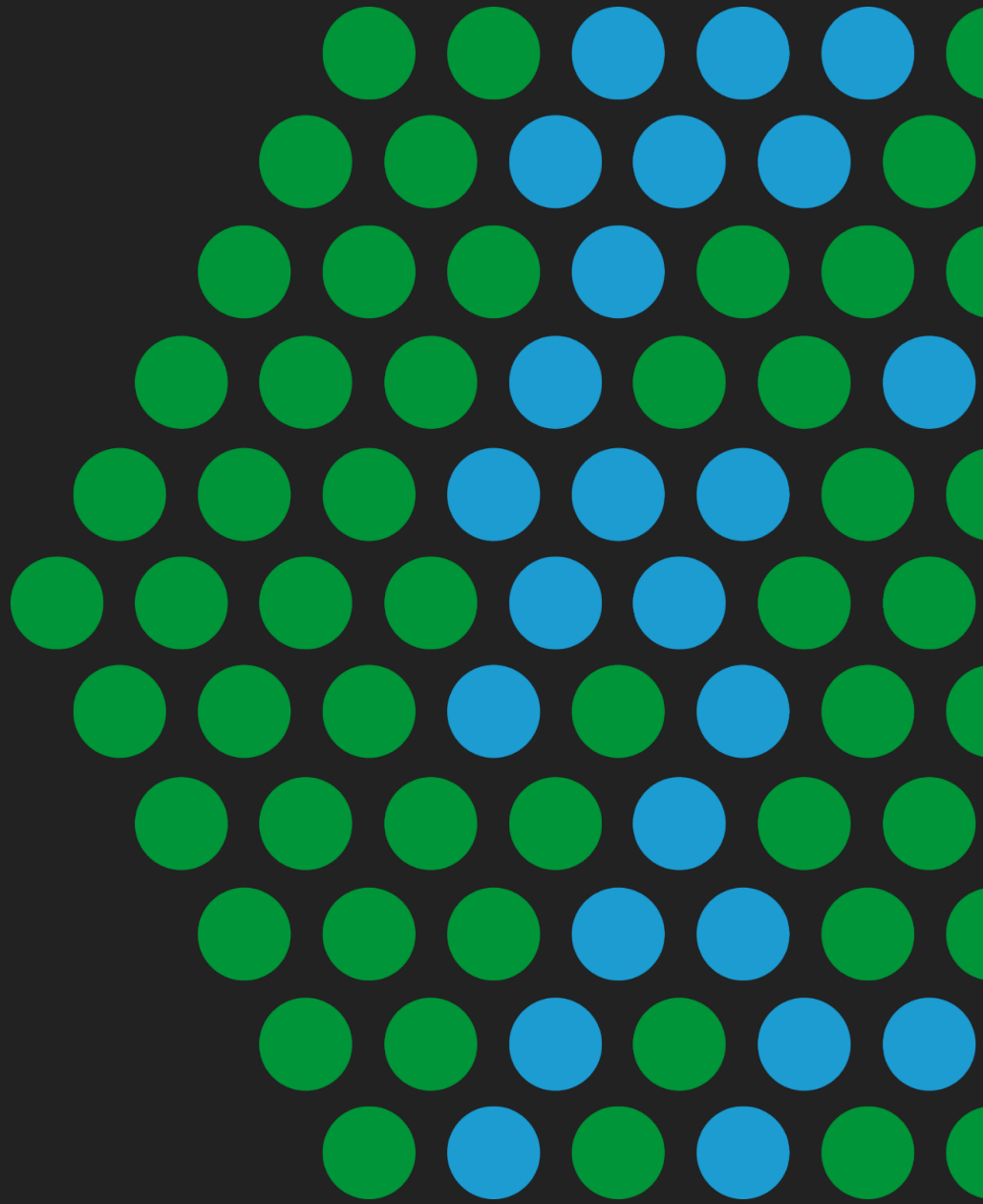
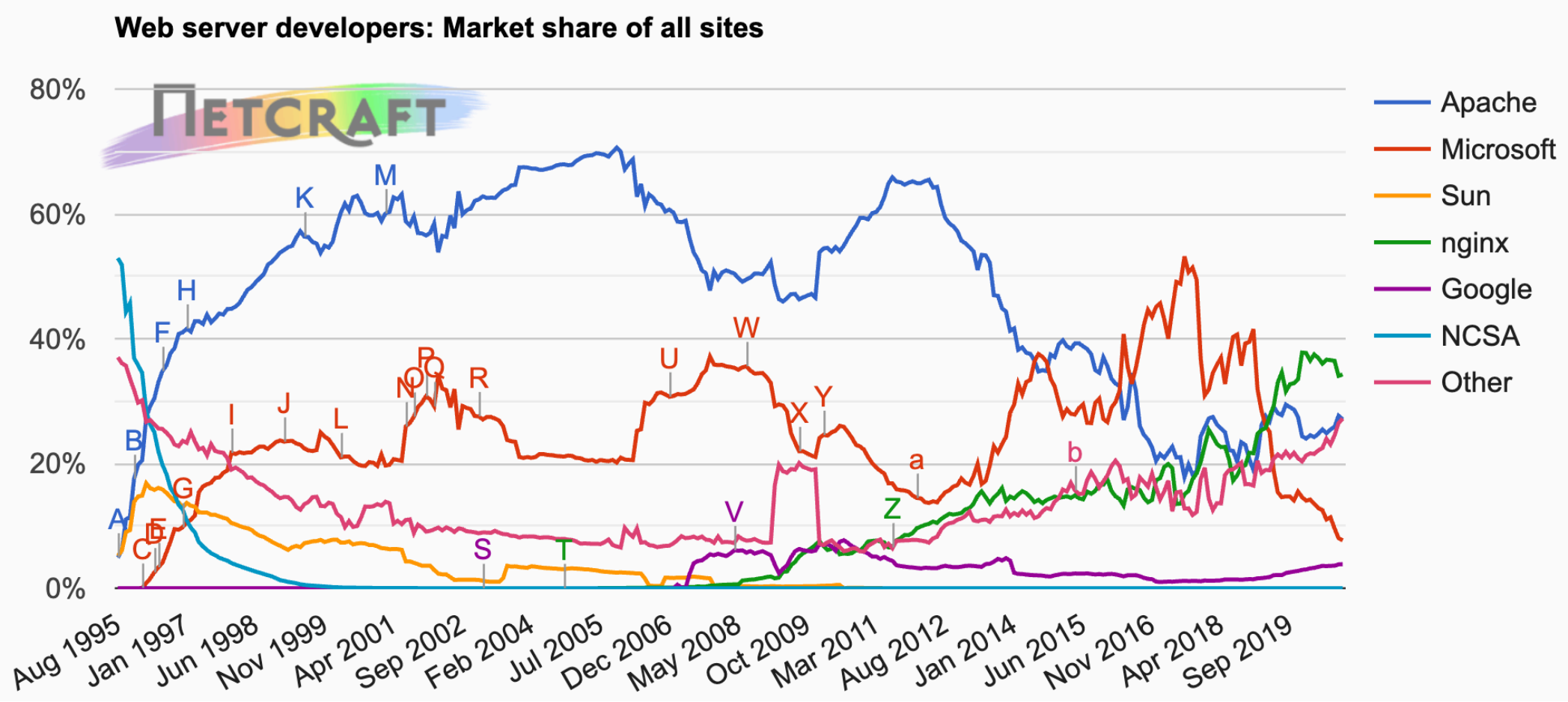
NGINX
Part of F5

# Automate your NGINX Controller AWS deployment in three simple steps

**Alessandro Fael Garcia (@alessfg)**
Technical Marketing Engineer

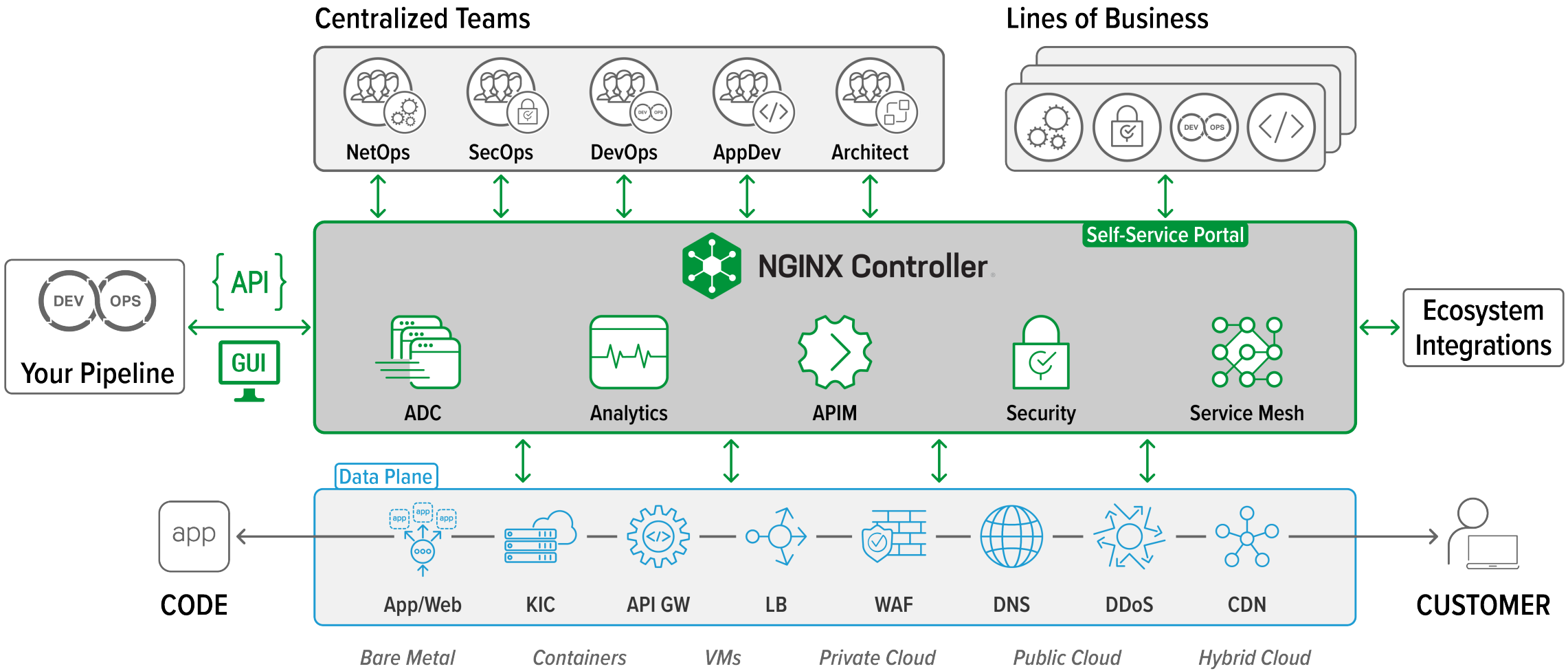


# What is NGINX?




HUBLOT
Santander
SINGHA
KASPERSKY
Santander
SCUDERIA FERRARI
7
V-Power
Santander
MAHLE
Ray-Ban
infor
Experis
SKF
NGK
brembo

# NGINX Controller



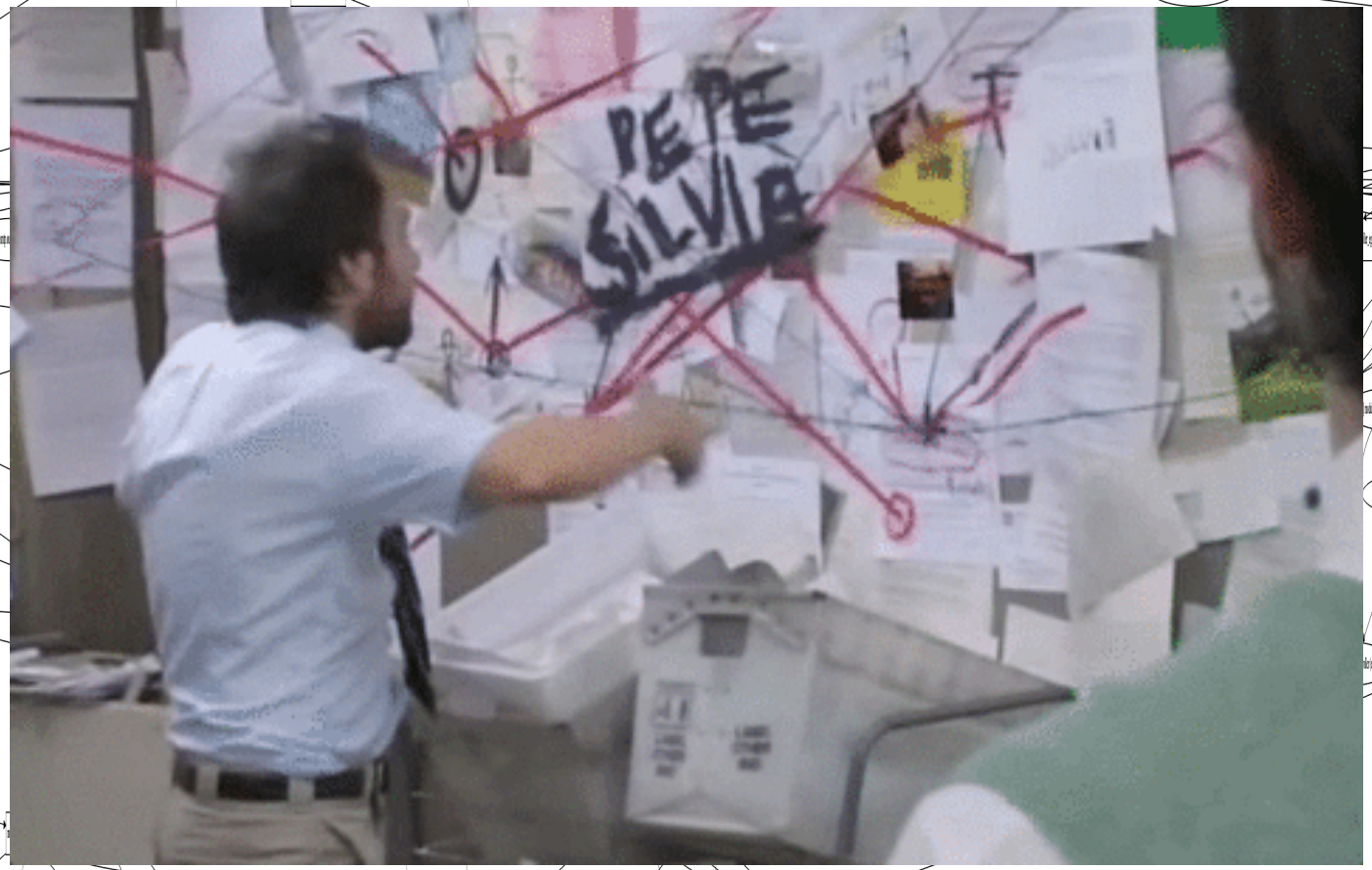
# NGINX Controller requirements

**Required:**

- PostgreSQL DB instance
  - v9.5/v12.x
  - User with CREATEDB permission
  - Can use bundled database in development environments (instance is recommended for production environments)
- NGINX Controller instance
  - NGINX Controller installation files
  - Package prerequisites

**Optional:**

- SMTP instance
  - Can redirect traffic to localhost in development environments (instance is recommended for production environments)
- NGINX instances
  - How to provision NGINX instances is left to the end user
  - Need to download/install the NGINX Controller agent (from the NGINX Controller instance) to connect instance to NGINX Controller


PE PE
SILVIA

# Automate your NGINX Controller AWS deployment

IN THREE STEPS!

1


HashiCorp Packer

"Pre-bake" NGINX Controller AMIs

2

HashiCorp Terraform

Deploy NGINX Controller infrastructure

3

ANSIBLE

Install and configure NGINX Controller and NGINX Controller agent

*Demo available at https://github.com/alessfg/nginx-controller-automation-demo*

# Thank you!

Get secure, scalable, and resilient application delivery with NGINX Controller!

nginx.com/nginx-controller/

twitter.com/nginx

youtube.com/user/nginxinc

linkedin.com/company/nginx

1

# HashiCorp Packer

- Create identical machine images (a.k.a. “pre-baked” VMs) on multiple environments from a single source configuration file
- Describe your machine image using HCL (Hashicorp Configuration Language)
- Native support for a wide range of configuration management providers such as Ansible

```
source "amazon-ebs" "nginx" {
  # AMI configuration
  ami_name              = "NGINX"
  ami_description       = "NGINX image"
  force_deregister      = true
  force_delete_snapshot = true
  ssh_username          = "ubuntu"
  tags = {
    Author = var.author
    Name   = "NGINX image"
    user   = "ubuntu"
  }
  # Access configuration
  region        = var.region
  # Run configuration
  instance_type = var.instance_type
  source_ami    = var.source_ami
}

build {
  description = "Install NGINX"
  sources = [
    "source.amazon-ebs.nginx"
  ]
  provisioner "ansible" {
    galaxy_file   = "${path.root}/requirements.yml"
    playbook_file = "${path.root}/nginx.yml"
    # Global parameters
    max_retries = 2
    pause_before = "10s"
  }
}
```

2

HashiCorp

# Terraform

- Deploy Infrastructure as Code
- Describe Infrastructure as Code using HCL (Hashicorp Configuration Language)
- Includes wide array of providers covering all major infrastructure platforms

```
resource "aws_instance" "debian_9" {
  count         = var.ami["debian_9"]["deploy"] ? 1 : 0
  ami           = var.ami["debian_9"]["ami_id"]
  instance_type = var.machine_type
  key_name      = var.key_name
  vpc_security_group_ids = [
    aws_security_group.main.id
  ]
  subnet_id = aws_subnet.main.id
  user_data = <<EOF
#! /bin/bash
apt update
apt install -y python3 python3-apt python-apt
EOF
  tags = {
    Name   = "debian_9"
    user   = "admin"
    author = var.key_name
    type   = "ansible"
  }
}
```

3


- Provisionment and configuration management in any environment
- Describe target host state using YAML
- Includes wide array of modules (written in Python) and roles (written in YAML) covering multiple use cases

```yaml
---
- hosts: aws_ec2
  become: true
  remote_user: "{{ hostvars[inventory_hostname].tags.user }}"
  collections:
    - nginxinc.nginx_core
  roles:
    - role: nginx
      vars:
        nginx_type: plus
        nginx_license:
          certificate: <path/to/cert>
          key: <path/to/key>
        nginx_remove_license: false
```