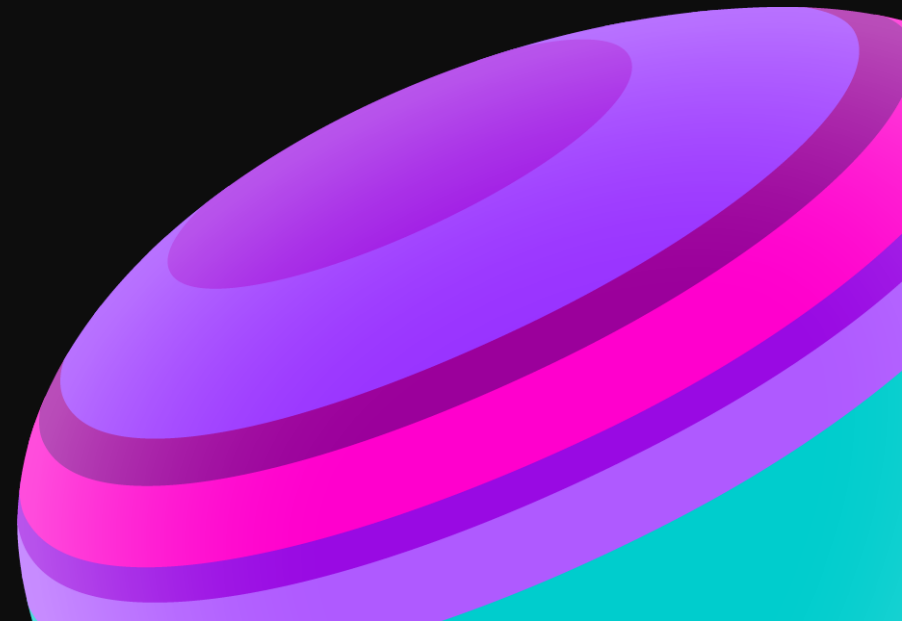




GRUNDLÄGGANDE AUTOMATION

TECHDAYS 2019

Christofer Tibbelin



JAG

- Namn: Christofer Tibbelin
- Yrke: Säkerhetsarkitekt
- Företag: Cygate
- Fritid: Spela spel (Minecraft SevTech Ages) + Segla
- Sport: Klättring



AGENDA

- Vad är automation.
- DevOps
- Skillnader
- Demon



VAD ÄR AUTOMATION?

- Göra något en gång istället för många.
- Förbereda och definiera vad som skall göras många gånger.
- Göra rätt och inte råka göra fel?
- Minska antal enkla och tråkiga saker.
- Att öka säkerhet och tillgänglighet.
- Ge bättre service till kunder.



DEVOPS VAD ÄR DET?

- (Utveckling och IT Drift)
- DevOps förbättrar relationen mellan Dev och Ops.
- Automation är en del av DevOps
- Agilt



PERIODIC TABLE OF DEVOPS TOOLS (V3) <https://xebialabs.com/periodic-table-of-devops-tools/>

Os Open Source
Fr Free
Fm Freemium
Pd Paid
En Enterprise

Source Control Mgmt.
Database Automation
Continuous Integration
Testing
Configuration

Deployment
Containers
Release Orchestration
Cloud
AIOps

Analytics
Monitoring
Security
Collaboration

PERIODIC TABLE OF DEVOPS TOOLS (V3) https://xebialabs.com/periodic-table-of-devops-tools/																							
1 GI GitLab																			2 Sp Splunk				
3 Gh GitHub	4 Dt Datical																	5 XLr XebiaLabs XL Release	6 Aws AWS	7 Az Azure	8 Gc Google Cloud	9 Op OpenShift	10 Sg Sumo Logic
11 Sv Subversion	12 Db DBMaestro																	13 Dk Docker	14 Ur UrbanCode Release	15 Af Azure Functions	16 Ld Lambda	17 Ic IBM Cloud	18 Fd Fluentd
19 Cw ISPW	20 Dp Delphix	21 Jn Jenkins	22 Cs Codeship	23 Fn FitNesse	24 Ju JUnit	25 Ka Karma	26 Su SoapUI	27 Ch Chef	28 Tf Terraform	29 XLd XebiaLabs XL Deploy	30 Ud UrbanCode Deploy	31 Ku Kubernetes	32 Cc CA CD Director	33 Pr Plutora Release	34 Al Alibaba Cloud	35 Os OpenStack	36 Ps Prometheus						
37 At Artifactory	38 Rg Redgate	39 Ba Bamboo	40 Vs VSTS	41 Se Selenium	42 Jm JMeter	43 Ja Jasmine	44 Sl Sauce Labs	45 An Ansible	46 Ru Rudder	47 Oc Octopus Deploy	48 Go GoCD	49 Ms Mesos	50 Gke GKE	51 Om OpenMake	52 Cp AWS CodePipeline	53 Cy Cloud Foundry	54 It ITRS						
55 Nx Nexus	56 Fw Flyway	57 Tr Travis CI	58 Tc TeamCity	59 Ga Gatling	60 Tn TestNG	61 Tt Tricentis Tosca	62 Pe Perfecto	63 Pu Puppet	64 Pa Packer	65 Cd AWS CodeDeploy	66 Ec ElectricCloud	67 Ra Rancher	68 Aks AKS	69 Rk Rkt	70 Sp Spinnaker	71 Ir Iron.io	72 Mg Moogsoft						
73 Bb BitBucket	74 Pf Perforce	75 Cr Circle CI	76 Cb AWS CodeBuild	77 Cu Cucumber	78 Mc Mocha	79 Lo Locust.io	80 Mf Micro Focus UFT	81 Sa Salt	82 Ce CFEngine	83 Eb ElasticBox	84 Ca CA Automic	85 De Docker Enterprise	86 Ae AWS ECS	87 Cf Codefresh	88 Hm Helm	89 Aw Apache OpenWhisk	90 Ls Logstash						



Follow @xebialabs

91 En XLi Xebialabs XL Impact	92 Os Ki Kibana	93 Fm Nr New Relic	94 En Dt Dynatrace	95 En Dd Datadog	96 Fm Ad AppDynamics	97 Os El ElasticSearch	98 Os Ni Nagios	99 Os Zb Zabbix	100 En Zn Zenoss	101 En Cx Checkmarx SAST	102 En Sg Signal Sciences	103 En Bd BlackDuck	104 Os Sr SonarQube	105 Os Hv HashiCorp Vault
106 En Sw ServiceNow	107 Pd Jr Jira	108 Fm Tl Trello	109 Fm Sk Slack	110 Fm St Stride	111 En Cn CollabNet VersionOne	112 En Ry Remedy	113 En Ac Agile Central	114 Pd Og OpsGenie	115 Pd Pd Pagerduty	116 Os Sn Snort	117 Fm Tw Tripwire	118 En Ck CyberArk	119 En Vc Veracode	120 En Ff Fortify SCA

VAD ÄR KONFIGURATIONS AUTOMATION

- Att alla enheter har rätt konfiguration
- Att de är rätt patchade
- Att samla in information om enheter
- Att man jobbar på samma sätt.
- Minska antal tråkiga repetitiva saker.
- Öka kollaboration mellan team.
- Spara tid och pengar



CHEF, PUPPET, SALTSTACK OCH ANSIBLE

- Mjukvara för att lösa konfigurationsautomation.
- Plus mycket annat.
- Är alla open source, men kan köpas också
- Skalbara från litet till stort.
- Är de jag hittade när jag googlade.
- Har vissa skillnader ändå.



SKILLNADER

	CHEF	Puppet	SaltStack	Ansible
Licens	OpenSource+	OpenSource+	OpenSource+	OpenSource+
Arkitektur	Master-Agent	Master-Agent	Master-Agent	Master-Klient
Agent på klienter?	Ja	Ja	Ja	Nej
Applicering	Hämta	Hämta	Skicka	Skicka
Installation	Medel svår	Medel svår	Medel svår	Lätt
Administration	Medel svår	Medel svår	Lätt	Lätt
Språk	DSL(Ruby)	DSL(PuppetDSL)	YAML(Python)	YAML(Python)



CHEF



- Klient till server arkitektur
- Imperativ programmeringsstil
- Skrivs i Ruby DSL
- Enkelt att göra debug på.
- Hämtar info från master server enligt schema. Ej direkt push
- Använder agenter som installeras på klienter
- Flexibel
- Svårare att lära sig
- Komplicerad



PUPPET



puppet

- Pionjär inom konfigurationsautomation
- Agent till master arkitektur
- Imperativ programmeringsstil
- Skrivs i Puppet DSL eller Ruby
- Hämtar info från master server enligt schema. Ej direkt push
- Använder agenter som installeras på klienter
- Lätt att fastna i en viss version
- Svårare att lära sig
- Komplicerad

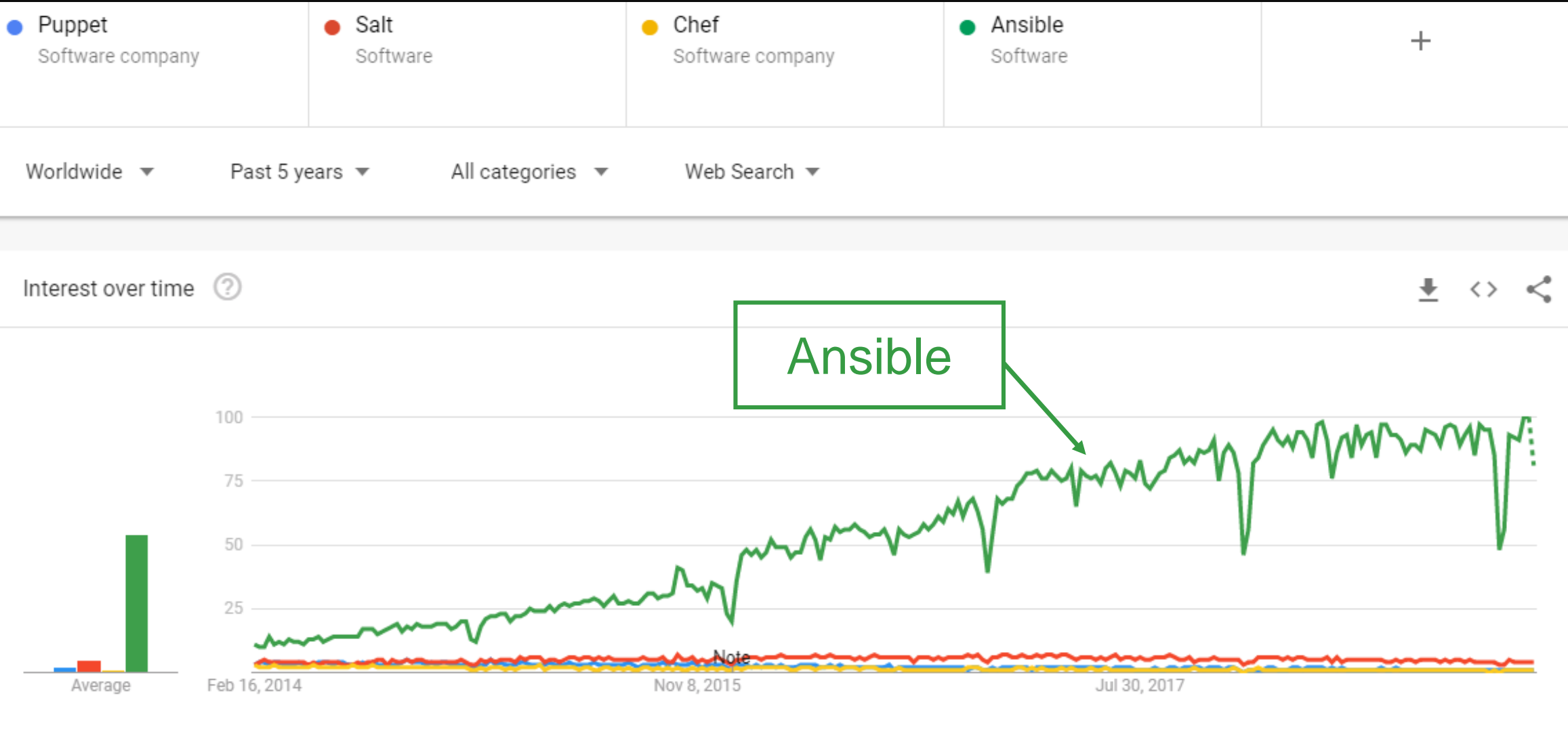


SALT (STACK)

- Låg-latency och hög hastighet
- Master till agent arkitektur via SSH
- Imperativ och deklarativ
- Skrivs i Python + YAML
- Push modell från master till klient
- Använder agenter som standard (svårt för switchar)
- Komplicerad installation
- Inte superbra dokumentation



GOOGLE TRENDS



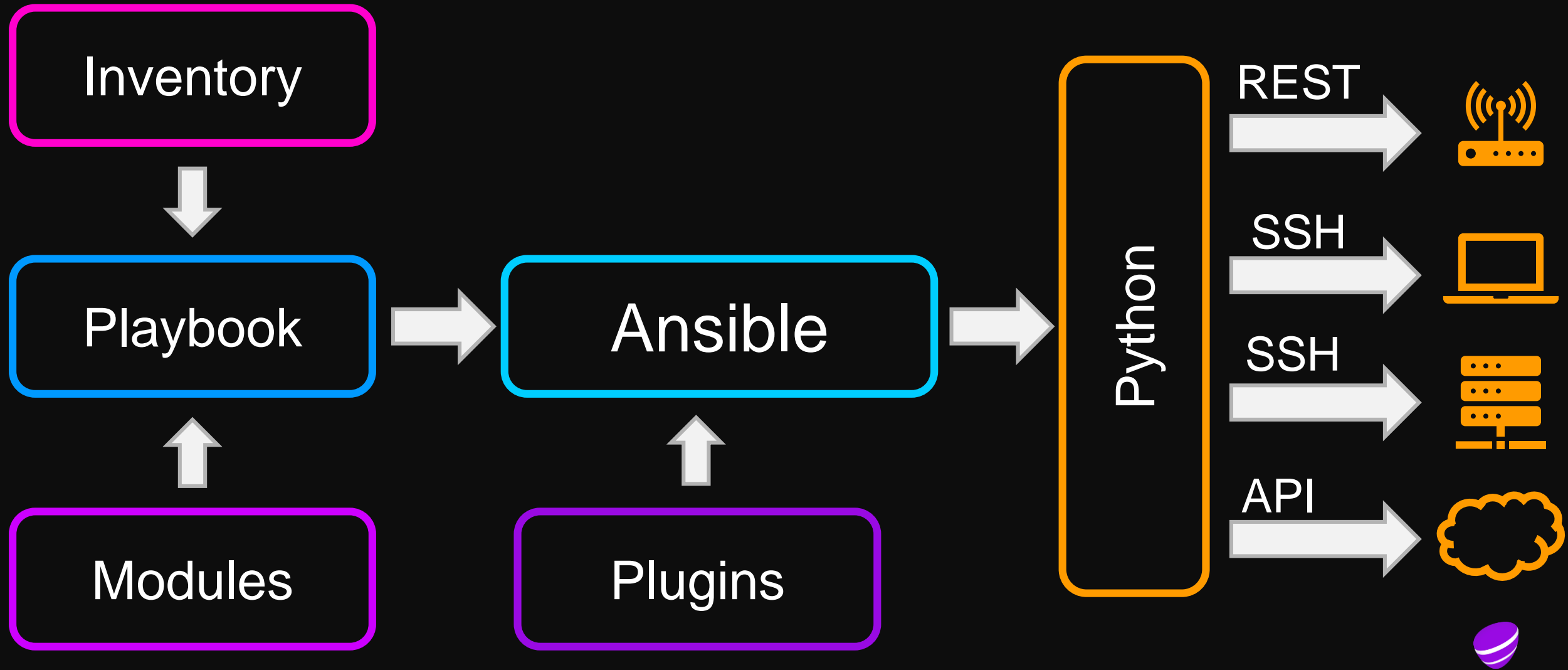


- Använder SSH * för kommunikation
- Hög säkerhet med SSH
- Konfiguration skrivs i YAML
- Använder inte agenter *
- Enkel installation och att använda
- Bra dokumentation
- Störst + bra stöd.





GRUNDER





- Hålla reda på mjukvarukod + och versioner till dessa.
- GitHub (publikt) eller GitLab (Privat)
- Enkelt för att jobba med rätt version och i team.
- ALLA använder GIT... är typ ett måste...
- Jag använder GitHub i Demo.
- <https://github.com/tibbe78/techday-2019>



YAML (*YAML AIN'T MARKUP LANGUAGE*)

variable_likes = “Being Lazy”

stoffe:

name: Christofer Tibbelin

age: 40

IQ: 42

likes:

- sweets**
- “{{ variable_likes }}”**



- DEMO!!

<https://github.com/tibbe78/techday-2019>



CYGATE TECHDAYS 2019 AUTOMATION DEMO MAP

My Laptop +
+ VirtualBox



Laptop
hostbridge
interface

Laptop
NAT out
interface



Internet

dmz1
Ubuntu Server

dmz2
Ubuntu Server

DMZ

OOB

Internet

Pfsense
Router & FW

MGT

cp_mgt1
CheckPoint MGT & FW

intranet

server1
Ubuntu Server

server2
Ubuntu Server

Ubuntu 18.10 Client

Router

Ansible

Docker

cp_api1

mgt_net

ANSIBLE PLAYBOOK

- Playbook används för att applicera en eller flera tasks (eller roles) till en eller flera serverar
- Ansible-playbook -K playbook.yml
- Every playbook starts with 3 hyphens '—'
- **Host section** – Defines the target machines on which the playbook should run. This is based on the Ansible inventory file.
- **Variable section** – This is optional and can declare all the variables needed in the playbook. We will look at some examples as well.
- **Tasks section** – This section lists out all the tasks that should be executed on the target machine. It specifies the use of Modules. Every task has a name which is a small description of what the task will do and will be listed while the playbook is run.



HUR ANVÄNDER JAG DETTA NU DÅ?

- Se till att all er utrustning har rätt version.
- Rulla ut ny mjukvara till servrar på ett standardiserat sätt.
- Skapa servrar och virtuella maskiner från start.
- Ordning och reda.
- Skapa en dokumentation som kan återskapas.
- Minska personberoende.
- Förbättra samarbete mellan olika team.



ANSIBLE BEST PRACTICE

- KISS
- Optimera för läsbarhet
- Tänk deklarativt
- Skapa en guidebook
- Versionshantera allt, GIT!!
- Starta litet och öka komplexitet sedan.
- Inventory ska ha mänskliga namn. Så man förstår.
- Prefix rollnamn in variabler.
- En variabel skall inte deklareras flera gånger i playbooks.



BEST PRACTICE

- Namnge allt i ansible.
- Ha inte för stora playbooks.
- Dela upp provision och configuration
- Använd inte shell eller command om man inte måste.
- Läs manualen...
- Debug bara när det behövs
- Starta inte bara en service. Kolla att den är uppe.
- Gör egna moduler om du får för mycket komplexitet
- Jinja2 template engine finns i Ansible.
- Templates ska vara enkla.



**SLUTA OROA ER
ÖVER VAD SOM KAN GÅ FEL**

**ISTÄLLET BLI EXALTERADE
ÖVER VAD SOM KAN GÅ RÄTT**

