How to find your way through the Automation Jungle

Eric Berg

Lead IT-Architect @ COMPAREX – Team Azure

Azure, Datacenter and Modern Workplace

Azure, System Center, Windows Server and Client

info@ericberg.de

@ericberg_de | @GeekZeugs

www.ericberg.de | www.geekzeugs.de

Microsoft® MVP Most Valuable Professional

GEEKZEUGS HOME OF

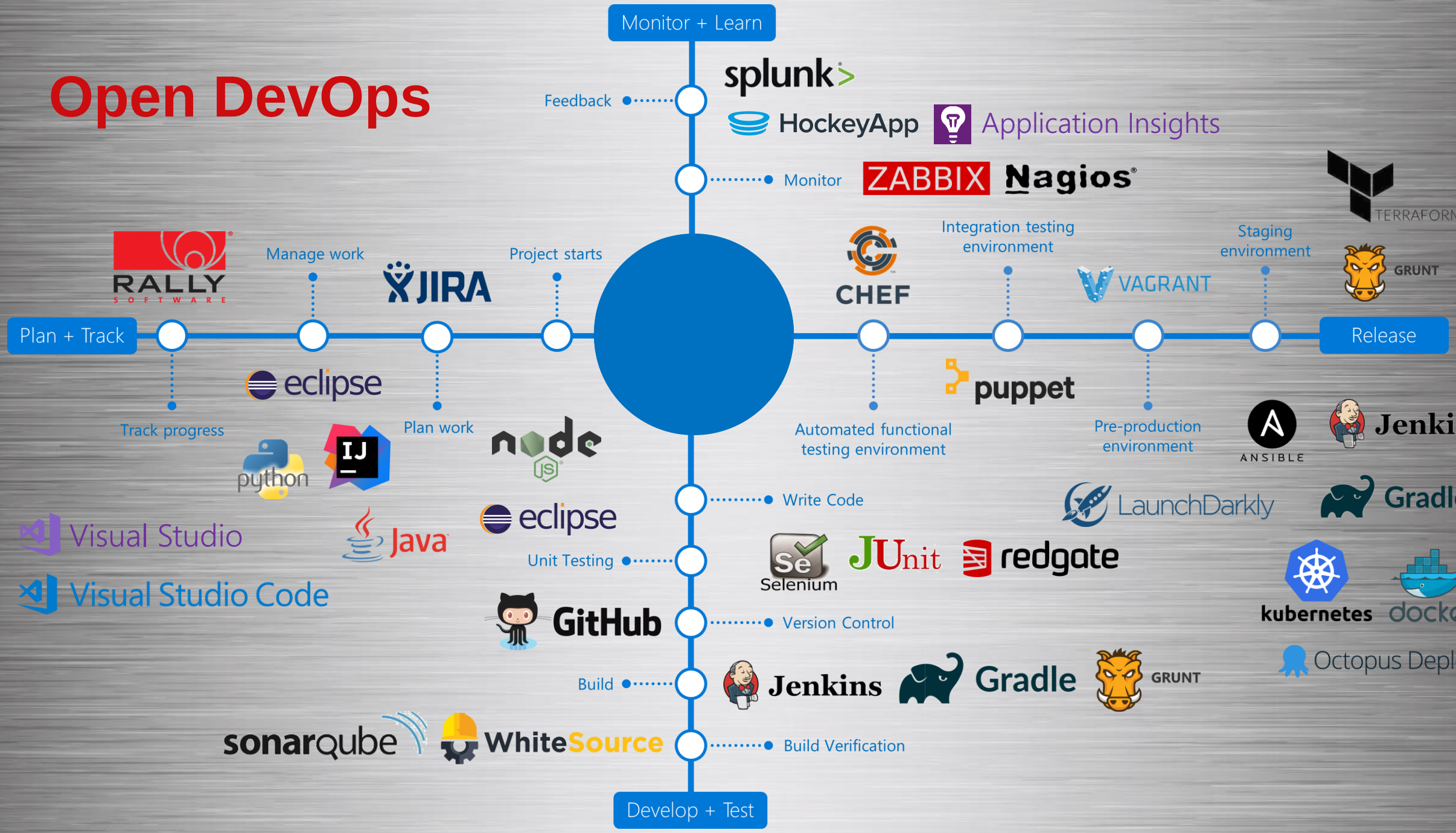
GEEKSPRECH Podcast

GEEKSCHAU Webcast

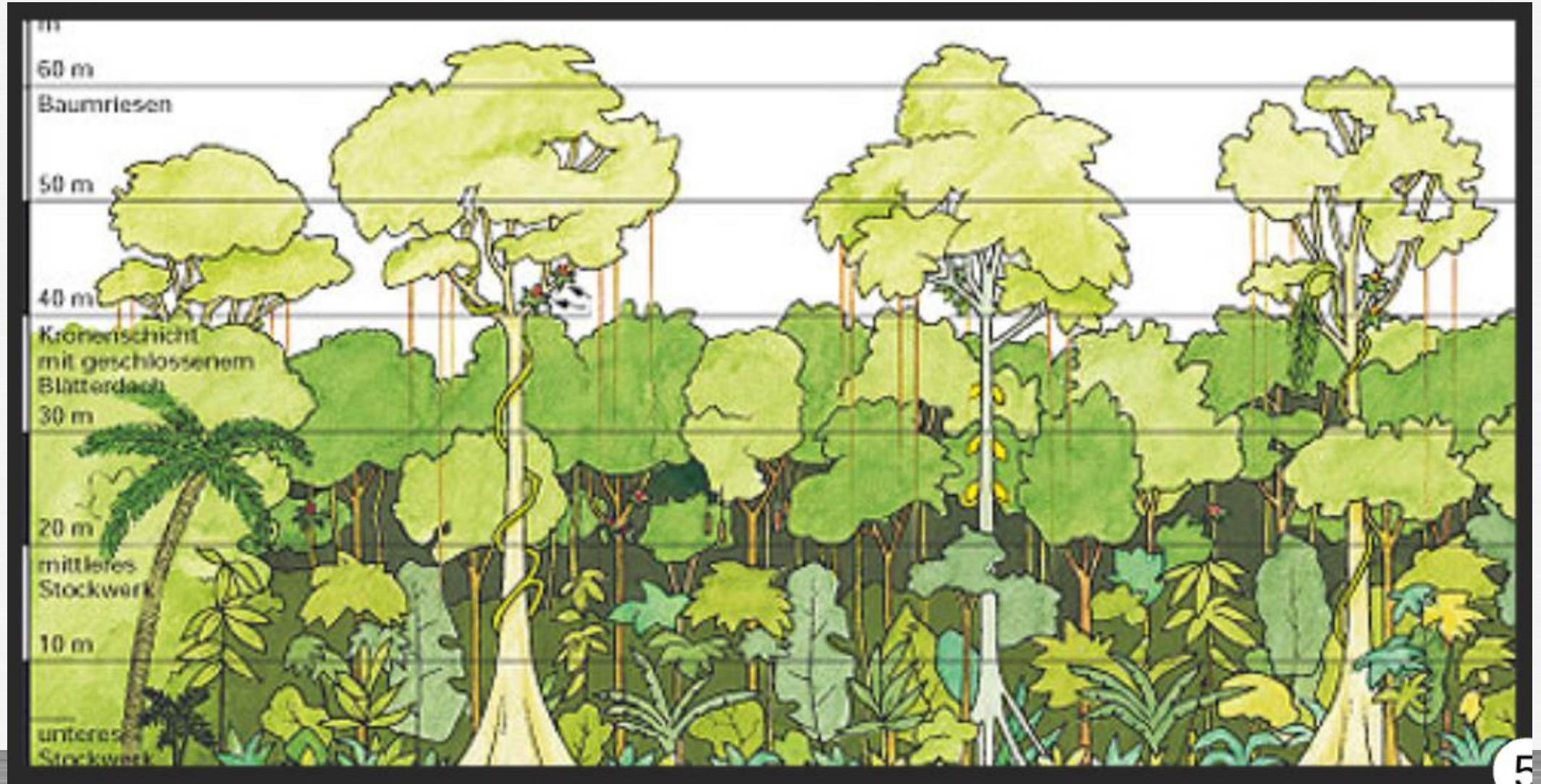
GEEKTREFF Community

Azure Meetup THÜRINGEN

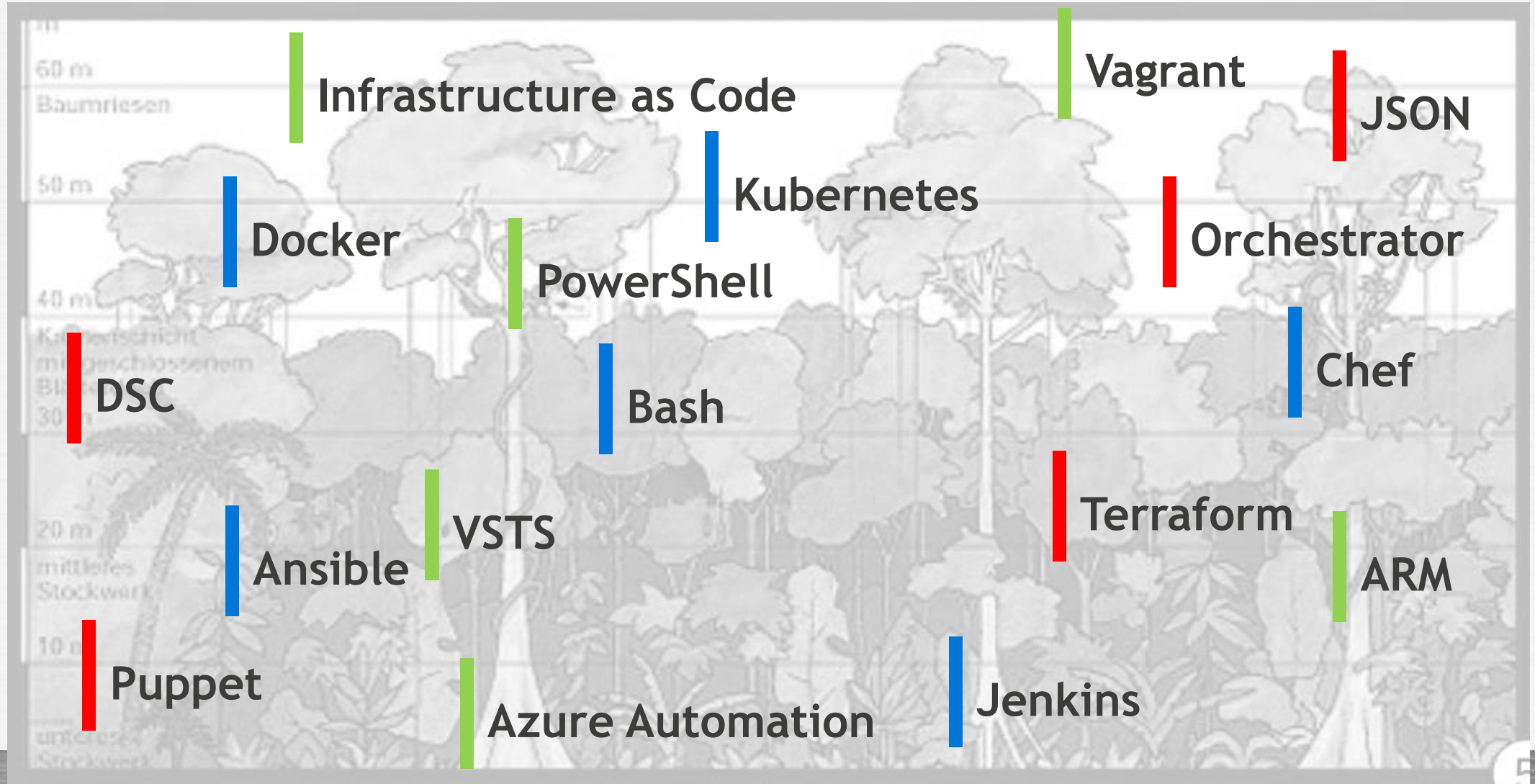
Open DevOps


Monitor + Learn
Feedback
splunk>
HockeyApp
Application Insights
Monitor
ZABBIX
Nagios
TERRAFORM
RALLY SOFTWARE
Manage work
JIRA
Project starts
CHEF
Integration testing environment
VAGRANT
Staging environment
GRUNT
Plan + Track
Release
eclipse
puppet
Track progress
Plan work
Automated functional testing environment
Pre-production environment
ANSIBLE
Jenkins
python
IJ
node
LaunchDarkly
Gradle
Write Code
Visual Studio
Java
eclipse
Unit Testing
Selenium
JUnit
redgate
kubernetes
docker
Visual Studio Code
GitHub
Version Control
Octopus Deploy
Build
Jenkins
Gradle
GRUNT
sonarqube
WhiteSource
Build Verification
Develop + Test

Jungle


60 m
Baumriesen
50 m
40 m
Kronenschicht
mit geschlossenem
Blätterdach
30 m
20 m
mittleres
Stockwerk
10 m

Automation Jungle

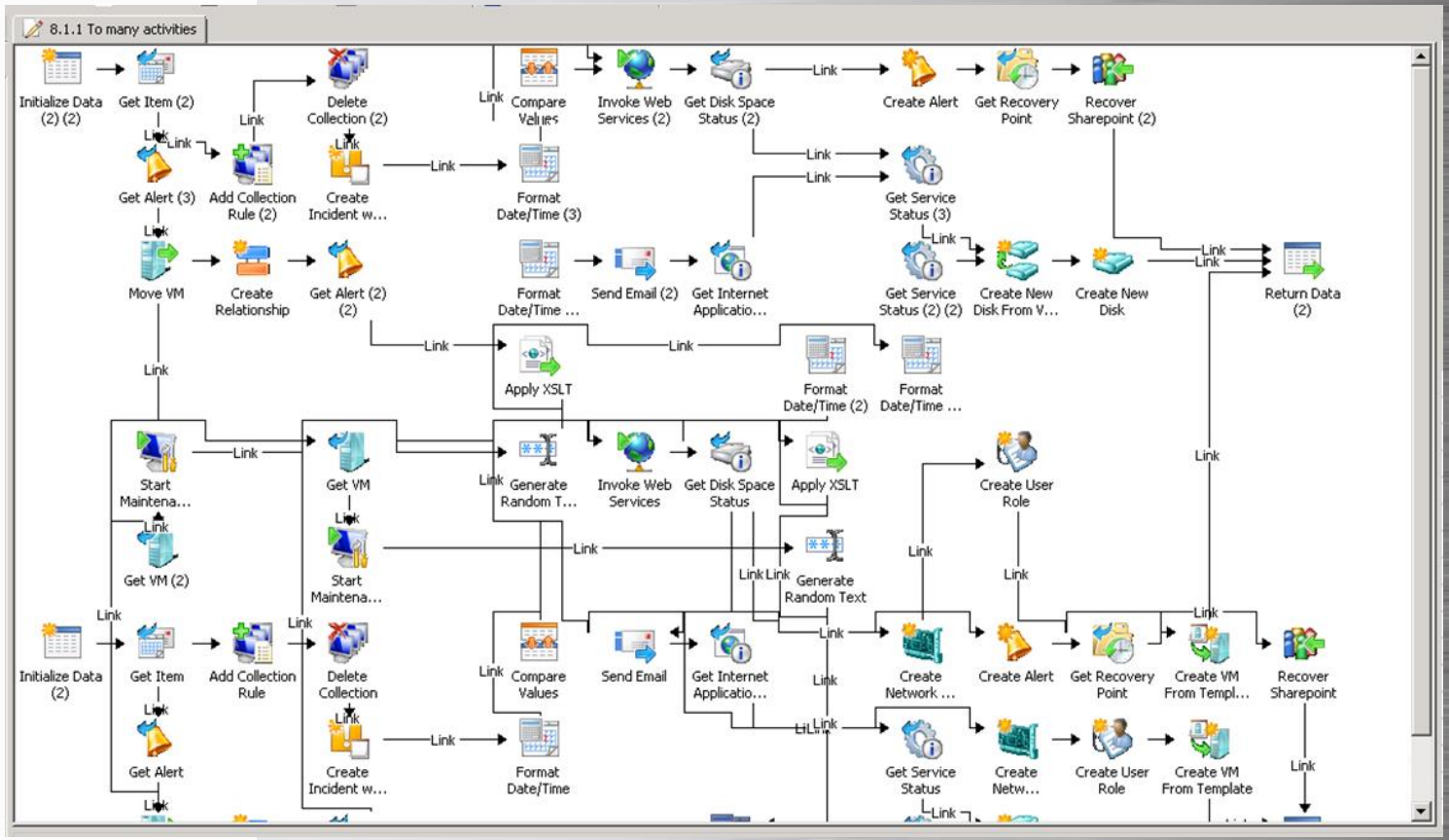


Automation - Facts

- FACT I
  - Automation often happens for the wrong reasons!
- FACT II
  - What does not exist cannot be automated!
- FACT III
  - A bad process does not get better when automated!
- FACT IV
  - Automation does not take up any jobs!

Automation - Pitfalls

- Pitfall I
  - The correct tool
- Pitfall 2
  - Everything at once
- Pitfall 3
  - Wrong priorization
- Pitfall 4
  - Missing standards
- Pitfall 5
  - Employees



Automation – Tools und Frameworks

Automation

Workflow Automation

- PowerShell
- Bash
- Au2mator
- Azure Automation

Configuration Management

- Puppet
- Chef
- Ansible
- DSC

Containerization

Virtualization

- Docker
- Kubernetes
- Vagrant

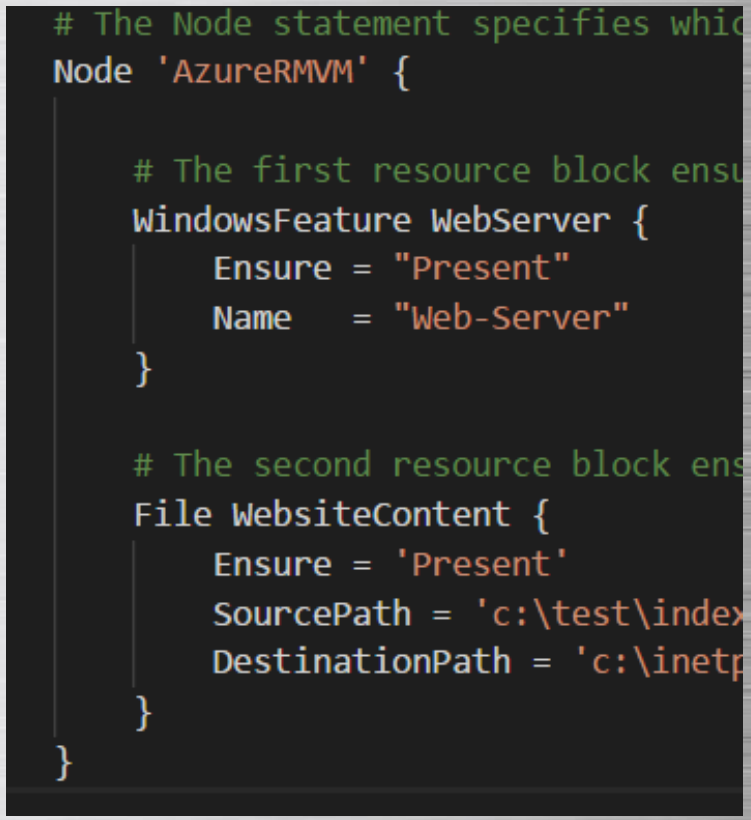
PowerShell

- Sequential processing
- Windows and Linux support
- Multiple cmdlets
- Error handling required
- Deployment + Configuration

```
#2. Check location
if(Check-AzureRmLocation -LocationName $Locatio
   #3. Check resource group, if not, created it
   if(Check-AzureRmResourceGroup -LocationName
      #4. Check VM images
      Write-Host "Check VM images $SkusName" -F
      If(Get-AzureRMVMImageSku -Location $Locat
         #5. Check VM
         If(Get-AzureRmVM -Name $VMName -Resou
            Write-Host -ForegroundColor Red "
         }
         else{
            #6. Check VM Size
            Write-Host "check VM Size $VMSizeN
            If(Get-AzureRmVMSize -Location $Lo
            {
               #7. Create a storage account
               $BlobURL = AutoGenerate-AzureRmS
               If($BlobURL){
                  #8. Create a network interface
```

PowerShell DSC

- Descriptive goal definition
- DSC providers define possibilities
- configuration management



```
# The Node statement specifies whic
Node 'AzureRMVM' {

    # The first resource block ensu
    WindowsFeature WebServer {
        Ensure = "Present"
        Name   = "Web-Server"
    }

    # The second resource block ens
    File WebsiteContent {
        Ensure = 'Present'
        SourcePath = 'c:\test\index
        DestinationPath = 'c:\inetp
    }
}
```

ARM / JSON

- Azure Resource Manager
- Descriptive goal definition
- Parallel Processing
- Extendable by extensions
- Natively available in Azure

```
"apiVersion": "2016-04-30-preview",
"type": "Microsoft.Compute/virtualMachines",
"name": "myVM",
"location": "[resourceGroup().location]",
"dependsOn": [
  "[resourceId('Microsoft.Network/networkInterfaces/', 'myNic')]"
],
"properties": {
  "hardwareProfile": { "vmSize": "Standard_DS1" },
  "osProfile": {
    "computerName": "myVM",
    "adminUsername": "[parameters('adminUsername')]",
    "adminPassword": "[parameters('adminPassword')]"
  },
  "storageProfile": {
    "imageReference": {
```

Terraform

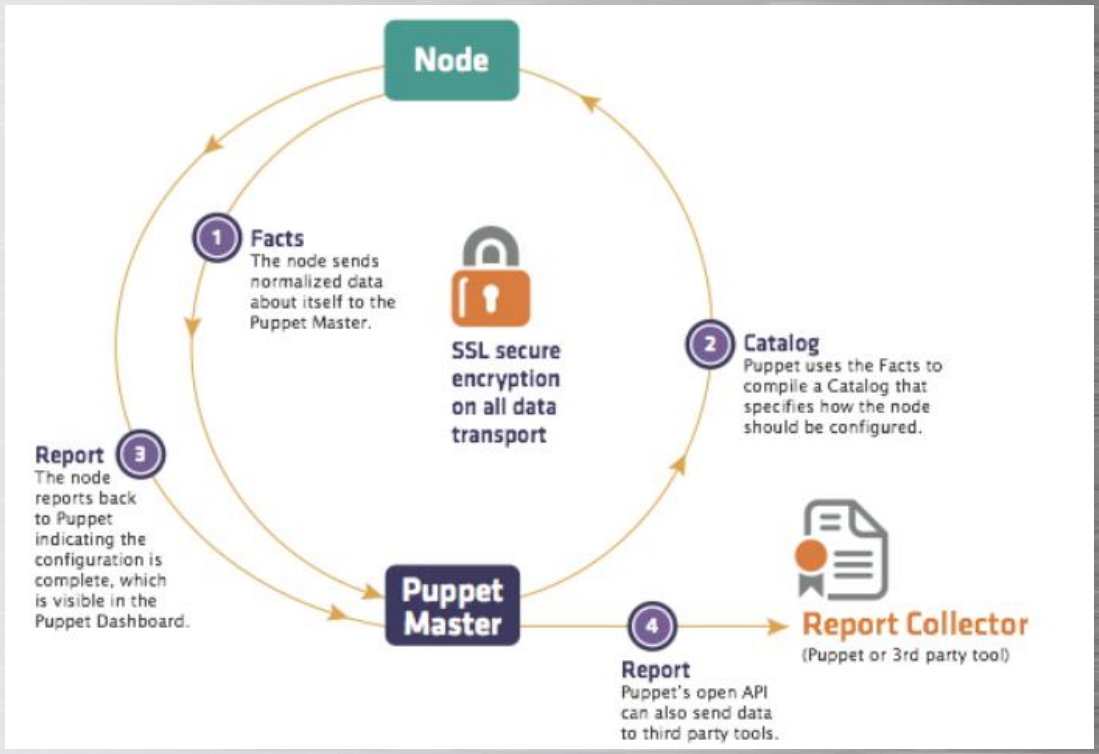
- Infrastructure as Code
- multi-cloud
- Integrated with Azure Cloud Shell
- No configuration management
- Integration with other tools possible

```
# create virtual machine
resource "azurerm_virtual_machine" "CDCTFVM" {
    name = "CDCTFVM"
    location = "West Europe"
    resource_group_name = "${azurerm_resource_group
    network_interface_ids = ["${azurerm_network_int
    vm_size = "Standard_A2"

    storage_image_reference {
        publisher = "Canonical"
        offer = "UbuntuServer"
        sku = "14.04.2-LTS"
        version = "latest"
    }
```

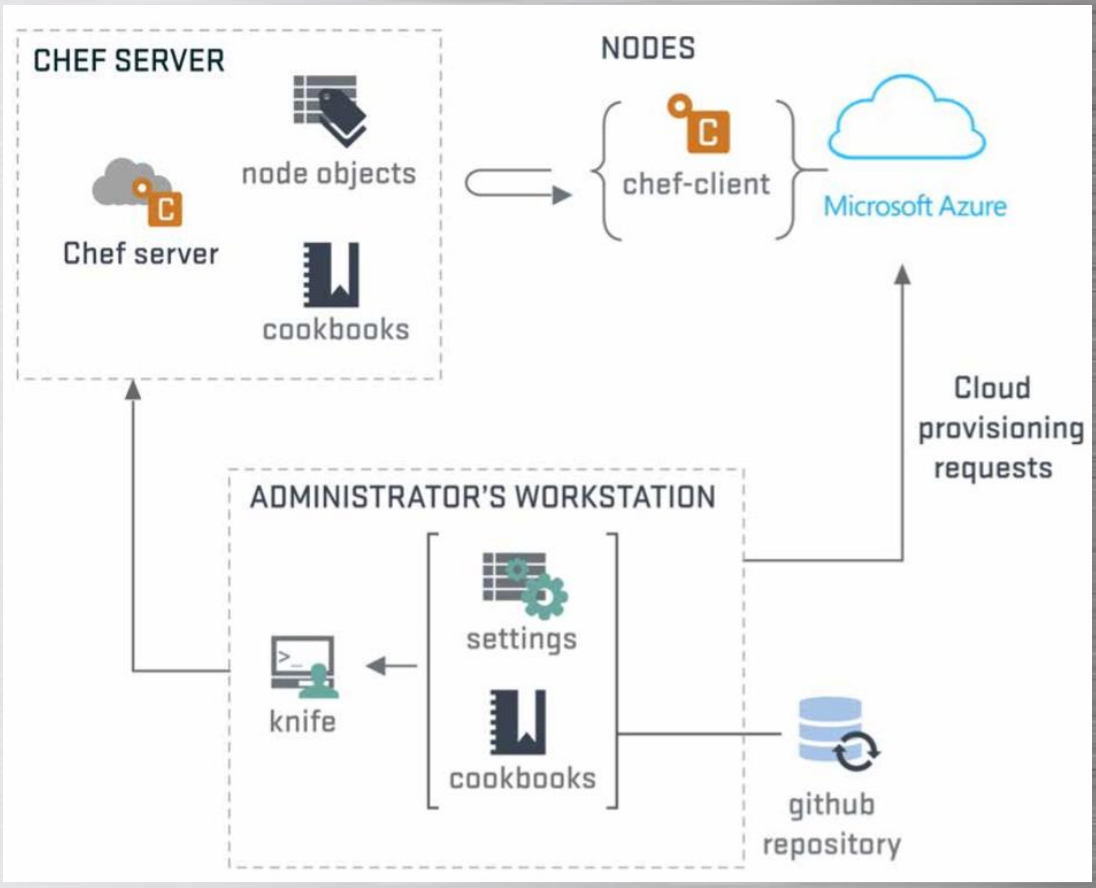
Puppet

- Agent based
- Pull Agents
- Configuration Management
- Facts vs. Catalog
- Desired State (Ruby)



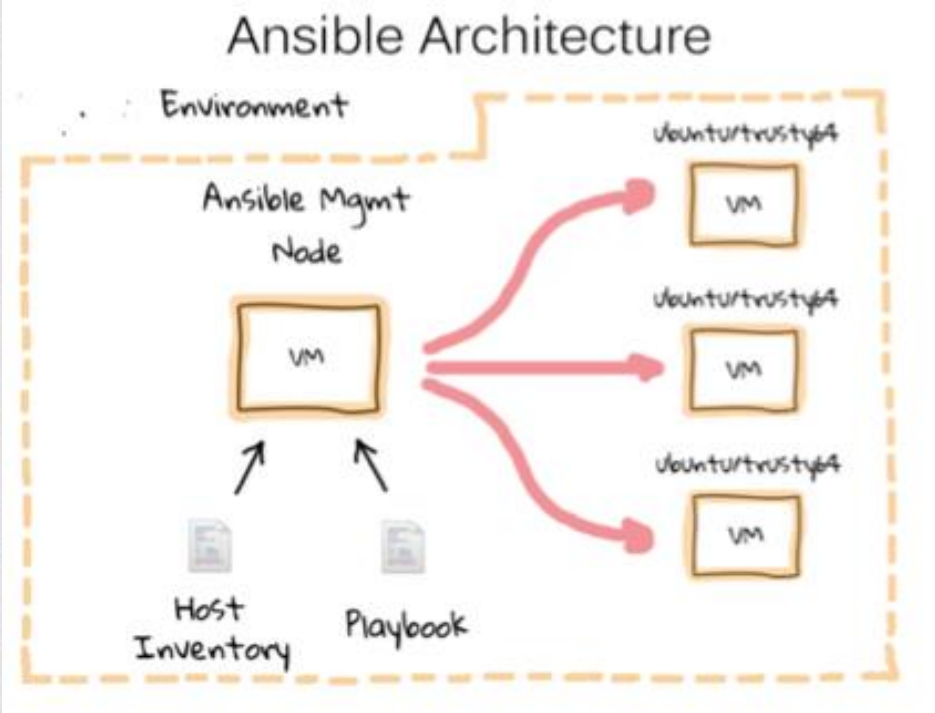
Chef

- Agent based
- Pull Agents
- Configuration Management
- Cookbooks (Ruby)
- Workflows

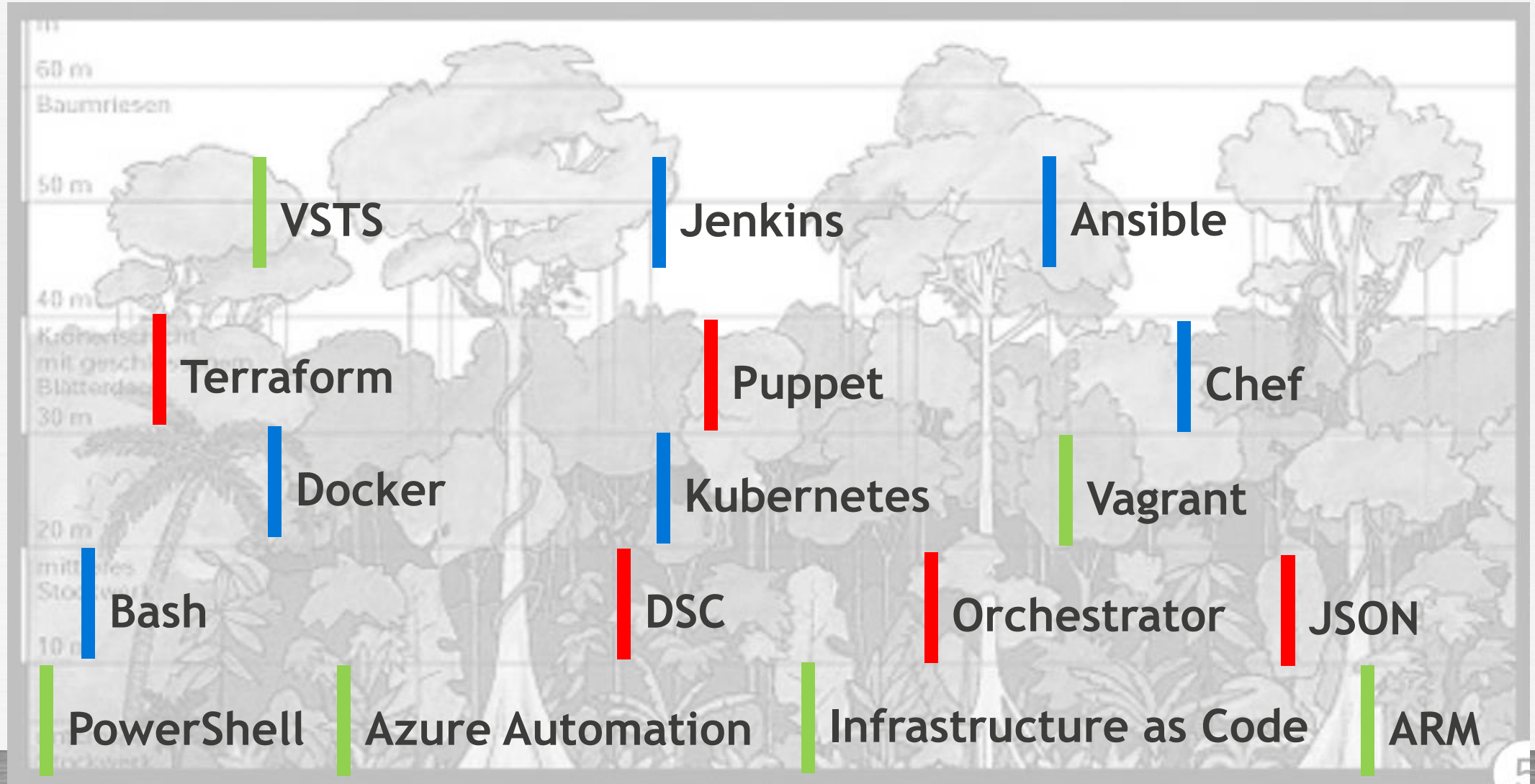


Ansible

- Integration into Azure Cloud Shell
- Agent-less
- Push based
- SSH or WinRM
- small / medium
- Phyton



Automation Jungle



Time for your questions…