

# ToiletPaper #113

## Machine images as code mit Packer

Autor: Philipp Hemmer / Senior Software Architect / Business Division Automotive Bavaria

### ✗ Problem

Container und Deployables in PaaS (Platform as a Service) Umgebungen sind gut und schön. Manchmal muss oder will man Maschinen-Images erstellen, z.B. für IaaS (Infrastructure as a Service) Umgebungen oder um oben genannte PaaS-Umgebungen aufzusetzen. Am besten noch unabhängig von der jeweiligen Virtualisierungsumgebung oder Cloud. Maschinen-Images haben auch den Vorteil, dass sie nur zum Erstellungszeitpunkt provisioniert werden müssen und danach schnell gestartet werden können.

Also wieder zurück zu selbstgestrickten Shell-Skripten und Handarbeit? Nein.

### ✓ Lösung

Hier gibt es ein Tool von HashiCorp, den Machern von Terraform, Vagrant, Vault etc.

**Packer** bietet die Möglichkeit, einmal das Maschinen-Image zu definieren und dann für verschiedene Virtualisierungslösungen und Clouds die Images zu erstellen. Man muss das Rad auch nicht neu erfinden, Packer unterstützt bekannte Provisionierungstools wie Chef, Ansible und mehr.

Bausteine von Packer:

- *Templates* sind die Konfigurationsdateien (JSON) von Packer und beschreiben ein oder mehrere Images.
- *Builders* produzieren die Images für jeweils eine Plattform.  
Z.B. Amazon EC2, Azure, Docker, Goggle Cloud, Vagrant, VirtualBox, VMWare, uvm.
- *Provisioners* installieren und konfigurieren das jeweilige Image.  
Z.B. mit Chef, Puppet, Ansible, Shell-Skripte, uvm.
- *Artifact* ist das Ergebnis eines Builders, das erzeugte Maschinen-Image.
- *Post-Processors* nimmt die erstellten Images und führt weitere Aufgaben aus (z.B. push in Docker Registry).

Mit folgendem einfachen Befehl werden alle definierten Images erzeugt:

```
packer build
```

### ➔ Beispiel

```
{
  "builders": [{
    "type": "amazon-ecs",
    "access_key": "{{user `aws_access_key`}}",
    "secret_key": "{{user `aws_secret_key`}}",
    "region": "us-east-1",
    "source_ami": "ami-43a15f3e",
    "instance_type": "t2.micro",
    "ssh_username": "ubuntu",
    "ami_name": "blend-reality-brenda-node-2.79b_{{timestamp}}",
    "vpc_id": "vpc-d52f5eb2",
    "subnet_id": "subnet-9942bbd0"
  }],
  "provisioners": [{
    "type": "ansible",
    "playbook_file": "ansible/playbook.yml"
  }]
}
```

Das Packer-Template erzeugt ein Amazon Maschine Image (AMI) in AWS. Das Image wird im Build mit Ansible provisioniert.

### + Weiterführende Aspekte

- <https://www.packer.io/>
- <https://github.com/kaorimat/packer-templates>
- [https://bitbucket.jambit.com/projects/POSTACK/repos/example\\_packer/browse](https://bitbucket.jambit.com/projects/POSTACK/repos/example_packer/browse)