

# Mastering Packer: A Guide to Automated Machine Image Creation

What is Packer?

**Packer** is an open-source DevOps tool developed by **HashiCorp**, designed to create identical machine images, popularly known as "golden images," which serve as blueprints for multiple platforms. It automates the creation of various machine images, including AWS AMIs, Azure machine templates, Docker containers, VMware machine templates, and more, ensuring consistent software configurations across all deployment environments.

## Key benefits include:

- **Automation**— Automates the image creation process, reducing manual effort and minimizing the risk of errors.
- **Immutability**— Promotes immutable infrastructure by replacing entire images instead of modifying existing ones.
- **Version Control**— Supports versioning and storage of images, ensuring a clear audit trail for deployments.
- **Integration**— Easily integrates with CI/CD pipelines, enhancing continuous integration and delivery workflow.

## Core concepts of Packer

To effectively utilize Packer, it is crucial to understand its key components:

- **Templates**: JSON files that define the configuration of Packer's components, outlining how machine images should be built. Templates include specifications for variables, builders and provisioners.
- **Packer** uses template-driven configuration files, written in JSON, to define the desired machine image specifications and build process.

```
{
  "variables": {
    "ssh_username": "ec2-user",
    "base_ami": "ami-0ba8711e0e1da2a52",
    "instance_type": "t2.micro"
  },

  "builders": [
    {
      "type": "amazon-ebs",
      "source_ami": "{{user `base_ami`}}",
      "instance_type": "{{user `instance_type` }}",
      "ssh_username": "{{user `ssh_username`}}",
      "ami_name": "packer-amazon-base2-{{timestamp}}"
    }
  ]
}
```

```

],

"provisioners": [
  {
    "type": "file",
    "source": "./files",
    "destination": "/tmp"
  },
  {
    "type": "shell",
    "script": "./scripts/install.sh"
  },
  {
    "type": "shell",
    "script": "./scripts/hardening.sh"
  }
]
}

```

The template is divided into three blocks:

1. **Variables:**

Custom variables that can be overridden during runtime using the `-var` flag. For example, in the snippet above, the AWS AMI name is specified as a variable.

2. **Builders:**

Defines the platforms for which images will be created. Multiple builders can be specified to target different platforms, such as EC2, Microsoft Azure, VMware, Google Cloud, Docker, and more.

3. **Provisioners:**

Allows you to provision the AMI by passing a shell script or using configuration management tools like Ansible, Chef, Puppet, or Shell. This step is used to install all the required packages and software.

Provisioners enhance your image by adding essential software and configurations, transforming a basic OS kernel image into a fully functional and ready-to-use machine image. Without provisioners, the image remains minimal and lacks the necessary components for practical use.

### Use Cases for Provisioners:

- Installing packages and software
- Creating and configuring user accounts
- Running custom scripts to make environment-specific changes
- Configuring storage and disk setups
- Implementing security configurations
- Downloading code files and other dependencies
- Patching the kernel
- Customizing and configuring software (e.g., Git, Docker)

In essence, provisioners play a crucial role in preparing the image, ensuring it is ready for deployment and use. Packer typically starts with a base Amazon Linux Image, known as the “Gold Image,” and applies these configurations to create a fully provisioned image.

### **Running Your First Packer Build Command**

Once Packer is installed and your environment is configured, you can run a basic build command to verify everything is set up correctly.

### **Step-by-Step Guide to Creating an Amazon Image Using Packer:**

1. Create a directory for your template.
2. Create a JSON configuration file.
3. Validate the configuration to ensure there are no errors.
4. Run the packer build command to start the image creation process.
5. Verify the newly created image to confirm it meets your requirements.

#### **1. Navigate to Packer Template Directory:**

#### **2. Validate the configuration**

```
packer validate simple-template.json
```

#### **3. Build the Image:**

Execute the following command:

```
packer build simple-template.json
```

Packer will initiate the image creation process based on your template specifications. The duration of the build will depend on its complexity.

#### 4. **Verify the Output:**

If the build is successful, Packer will display details about the created image, such as its ID and storage location.

#### **Verify the AMI Via AWS Console:**

- Navigate to the EC2 Dashboard.
- Go to Images → AMIs.
- Search for your AMI by its name or ID. Verify the status is available.
- Confirm that the AMI's status is available.

#### **Verify the AMI Via CLI: Run the following command:**

```
aws ec2 describe-images --owners self --query 'Images[*].[ImageId,Name,State]' --output table
```