Scenario 1

```
{
    "variables": {
        "aws_access_key": "",
        "aws_secret_key": ""
    },
    "builders": [
        {
            "type": "amazon-ebs",
            "access_key": "{{user `aws_access_key`}}",
            "secret_key": "{{user `aws_secret_key`}}",
            "region": "us-east-2",
            "source_ami_filter": {
                "filters": {
                    "virtualization-type": "hvm",
                    "name": "ubuntu/images/*ubuntu-xenial-16.04-amd64-server-*",
                    "root-device-type": "ebs"
                },
                "owners": [
                    "099720109477"
                ],
                "most_recent": true
            },
            "instance_type": "t2.micro",
            "ssh_username": "ubuntu",
            "ami_name": "scenario1-packer-{{timestamp}}"
        }
    ]
}
```

Le fichier de configuration JSON pour Packer présenté ici est conçu pour créer une image Amazon Machine (AMI) sur AWS (Amazon Web Services). Voici une explication détaillée des différentes parties de cette configuration :

Variables

```
jsonCopy code
"variables": {
    "aws_access_key": "",
    "aws_secret_key": ""
```

```
}
```

Cette section définit les variables qui seront utilisées dans la configuration. Les clés d'accès AWS (`aws_access_key` et `aws_secret_key`) sont laissées vides, ce qui signifie que vous devrez les fournir lors de l'exécution de la commande `packer build`.

Builders

```
jsonCopy code
"builders": [
    {
        "type": "amazon-ebs",
        "access_key": "{{user `aws_access_key`}}",
        "secret_key": "{{user `aws_secret_key`}}",
        ...
    }
]
```

Cette section contient les instructions pour le "builder", qui dans ce cas est de type `amazon-ebs`. Le builder utilise les variables `aws_access_key` et `aws_secret_key` pour se connecter à AWS.

Paramètres du Builder

- `region` **:** La région AWS où l'image sera créée (`us-east-2` dans cet exemple).
- `source_ami_filter` **:** Un filtre pour sélectionner l'AMI source à partir de laquelle créer la nouvelle AMI.
  - `filters` **:** Les critères de filtre spécifiques comme le type de virtualisation (`hvm`), le nom de l'AMI source (dans ce cas, une AMI Ubuntu Xenial 16.04), et le type de périphérique racine (`ebs`).
  - `owners` **:** L'ID du compte AWS propriétaire de l'AMI source.
  - `most_recent` **:** Utilise la plus récente des AMIs qui correspondent aux critères de filtre.
- `instance_type` **:** Le type d'instance EC2 à utiliser pour créer l'AMI (`t2.micro` dans cet exemple).

- `ssh_username` : Le nom d'utilisateur pour se connecter à l'instance via SSH (dans ce cas, `ubuntu`).
- `ami_name` : Le nom de la nouvelle AMI, avec un timestamp pour la rendre unique (`scenario1-packer-{{timestamp}}`).

Ce fichier de configuration utilise les fonctionnalités clés de Packer pour créer une AMI personnalisée dans AWS. Vous pouvez exécuter cette configuration avec la commande `packer build`, après avoir fourni les clés d'accès AWS appropriées.

Execution

Pour exécuter le flux de construction (build flow) de Packer avec ce fichier JSON appelé `scenario1.json`, voici les commandes que vous pouvez utiliser :

1. Valider la Configuration

Avant d'exécuter la construction, il est conseillé de valider le fichier de configuration pour vérifier qu'il n'y a pas d'erreurs.

```
packer validate scenario1.json
```

2. Définir les Variables d'Environnement (Optionnel)

Vous pouvez définir les variables d'environnement AWS pour ne pas avoir à les insérer manuellement. Remplacez `YOUR_AWS_ACCESS_KEY` et `YOUR_AWS_SECRET_KEY` par vos clés d'accès AWS.

```
export AWS_ACCESS_KEY_ID=YOUR_AWS_ACCESS_KEY
export AWS_SECRET_ACCESS_KEY=YOUR_AWS_SECRET_KEY
```

3. Exécuter la Construction

Pour démarrer le processus de création de l'AMI, exécutez la commande suivante :

```
packer build scenario1.json
```

Si vous n'avez pas défini les variables d'environnement, vous pouvez également passer les clés d'accès en tant que variables utilisateur directement dans la commande `packer build` comme suit :

```
packer build -var 'aws_access_key=YOUR_AWS_ACCESS_KEY' -var 'aws_secret_key=YOUR_AWS_SECRE
T_KEY' scenario1.json
```

Après avoir exécuté ces commandes, Packer va démarrer une instance EC2 temporaire dans AWS selon les spécifications du fichier `scenario1.json`, installer et configurer tout logiciel (si des "provisioners" sont spécifiés, ce qui n'est pas le cas dans l'exemple donné), créer une nouvelle AMI à partir de cette instance, puis terminer l'instance temporaire.

Une fois que le processus est terminé, vous aurez une nouvelle AMI prête à être utilisée, que vous pouvez ensuite déployer à l'aide de Terraform ou d'autres outils de gestion d'infrastructure.