

```

pipeline {
  agent { label 'build-server-1' } // Ensure job runs on the right node

  parameters {
    string(name: 'INSTALLER_URL', description: 'Enter the full path to the installer file')
    string(name: 'OWNER', description: 'Enter the Owner name')
    string(name: 'FME_BUILD', description: 'Enter the FME Build version')
    string(name: 'SOURCE_AMI', description: 'Enter the Source AMI ID (e.g.,
ami-xxxxxxxxxxxxxxxxxxx)')
  }

  environment {
    AWS_REGION = 'ca-central-1'
  }

  stages {
    stage('Validate User Inputs') {
      steps {
        script {
          if (!params.INSTALLER_URL?.trim()) {
            error "INSTALLER_URL cannot be empty. Please provide a valid S3 path."
          }
          if (!params.OWNER?.trim()) {
            error "OWNER cannot be empty. Please provide an Owner name."
          }
          if (!params.FME_BUILD?.trim()) {
            error "FME_BUILD cannot be empty. Please provide an FME build version."
          }
          if (!params.SOURCE_AMI?.trim() || !params.SOURCE_AMI.startsWith("ami-")) {
            error "SOURCE_AMI cannot be empty and must be a valid AMI ID (e.g.,
ami-xxxxxxxxxxxxxxxxxxx)."
          }
        }
      }
    }

    stage('Checkout Code') {
      steps {
        git branch: 'main', url: 'https://github.com/<Github Repo>/fme-server-iac-templates.git'
      }
    }

    stage('Initialize Packer') {
      steps {

```

```

    dir('AWS/packer') {
        sh 'packer init fme_flow_aws.pkr.hcl'
    }
}

stage('Validate Packer Template') {
    steps {
        dir('AWS/packer') {
            sh """
            packer validate \
            -var "region=$AWS_REGION" \
            -var "installer_url=${params.INSTALLER_URL}" \
            -var
            "tags={Owner=\\\"${params.OWNER}\\\",fme_build=\\\"${params.FME_BUILD}\\\"}" \
            -var "source_ami=${params.SOURCE_AMI}" \
            fme_flow_aws.pkr.hcl
            """
        }
    }
}

stage('Build AMI with Packer') {
    steps {
        dir('AWS/packer') {
            sh """
            packer build \
            -var "region=$AWS_REGION" \
            -var "installer_url=${params.INSTALLER_URL}" \
            -var
            "tags={Owner=\\\"${params.OWNER}\\\",fme_build=\\\"${params.FME_BUILD}\\\"}" \
            -var "source_ami=${params.SOURCE_AMI}" \
            fme_flow_aws.pkr.hcl | tee output.log
            """
        }
    }
}

stage('Extract AMI ID') {
    steps {
        dir('AWS/packer') {
            script {
                def ami_id = sh(script: "grep -o 'ami-[a-zA-Z0-9]*' output.log | tail -1",
returnStdout: true).trim()
                if (!ami_id) {
                    error "Failed to extract AMI ID. Please check the Packer logs."
                }
            }
        }
    }
}

```

```
    writeFile file: 'ami_id.txt', text: ami_id
    echo "AMI ID: ${ami_id}"
  }}
}
}
```