

Only you can see this message



This story's distribution setting is on. You're in the Partner Program, so this story is eligible to earn money. [Learn more](#)

# Migrating an app to AWS without the use of a container — Utilizing Elastic Beanstalk Custom Platforms.



Craig Godden-Payne

Jun 10 · 4 min read ★



Containers are awesome.

Being able to spin up an ephemeral copy of an application feels powerful.

Saving costs by significantly reducing CPU, memory and storage is phenomenal

The thing is, not all applications can be containerised, but there are options...

This post is going to look specifically at using custom platforms with Elastic Beanstalk.

*AWS Elastic Beanstalk supports custom platforms. A custom platform is a more advanced customization than a custom image in several ways. A custom platform lets you develop an entire new platform from scratch, customizing the operating system, additional software, and scripts that Elastic Beanstalk runs on platform instances*

### **The Only Step You Need to Progress in Your Career as a Developer or Engineer.**

I don't usually like to write articles about myself, as it feels a bit self-indulgent, but I thought it would be useful...

medium.com



Simply because elastic beanstalk does not provide a platform you require out of the



box, you should not be put off.

You can still use a language, framework or some infrastructure that you require, and manage it transiently.

Beanstalk custom platforms are comparable to a custom image, where you modify an existing base AMI.

You may take something like Ubuntu, and install your software on the image.

Elastic Beanstalk can still provide hooks for scripts and can control the overall stack.

The Elastic Beanstalk platform can be used in conjunction with Packer, (an open-source tool for creating machine images).

### **Creating a Simple, Low-Cost Twitter Bot, utilising Serverless Technologies.**

People have a love-hate relationship with twitter bots.

[medium.com](https://medium.com/@craig.beardy.digital/migrating-an-app-to-aws-without-the-use-of-a-container-using-elastic-beanstalk-custom-platforms-4f4bd5ed1e59)



Below is an example of running NodeJs on Ubuntu, built with packer for Elastic Beanstalk custom platform:

```
version: "1.0"

provisioner:
  type: packer
  template: custom_platform.json
  flavor: ubuntu1604

metadata:
  description: Sample NodeJs Container.
  operating_system_name: Ubuntu
  operating_system_version: 16.04
  programming_language_name: ECMAScript
  programming_language_version: ECMA-262
  framework_name: NodeJs
  framework_version: 13.0.1
  app_server_name: "none"
  app_server_version: "none"

option_definitions:
  - namespace: "aws:elasticbeanstalk:container:custom:application"
    option_name: "NPM_START"
    description: "Default application startup command"
    default_value: "node application.js"
```

Example of the custom platform template:

```
{
  "variables": {
    "platform_name": "{{env `AWS_EB_PLATFORM_NAME`}}",
    "platform_version": "{{env `AWS_EB_PLATFORM_VERSION`}}",
    "platform_arn": "{{env `AWS_EB_PLATFORM_ARN`}}"
  },
  "builders": [
    {
      "type": "amazon-ebs",
      "name": "HVM AMI builder",
      "region": "",
      "source_ami": "",
      "instance_type": "m3.medium",
      "ssh_username": "ubuntu",
      "ssh_ptty": "true",
      "ami_name": "Beanstalk Custom Platform - Node on Ubuntu",
      "tags": {
        "eb_platform_name": "{{user `platform_name`}}",
        "eb_platform_version": "{{user `platform_version`}}",

```

```

    "eb_platform_arn": "{{user `platform_arn`}}"
  }
}
],
"provisioners": [
  {
    "type": "file",
    "source": "builder",
    "destination": "/tmp/"
  },
  {
    "type": "shell",
    "execute_command": "chmod +x {{ .Path }}; {{ .Vars }} sudo {{
.Path }}",
    "scripts": [
      "builder/builder.sh"
    ]
  }
]
}
}

```

### Using Microservice Patterns in an increasingly Serverless world

When working on an application domain, it is beneficial to use the microservices software design pattern.

[medium.com](https://medium.com/@craig.beardy.digital/migrating-an-app-to-aws-without-the-use-of-a-container-using-elastic-beanstalk-custom-platforms-4f4bd5ed1e59)

Example builder.sh script:

```

BUILDER_DIR="/tmp/builder"

. $BUILDER_DIR/CONFIG

wait_for_cloudinit() {
    echo "Waiting for cloud init to finish bootstrapping.."
    until [ -f /var/lib/cloud/instance/boot-finished ]; do
        echo "Still bootstrapping.. sleeping. "
        sleep 3;
    done
}

run_command () {
    echo "Running script [$1]"
    chmod +x $1
    (cd $BUILDER_DIR/setup-scripts; BUILDER_DIR=$BUILDER_DIR $1 )
    if [ $? -ne "0" ]; then
        echo "Exiting. Failed to execute [$1]"
        exit 1
    fi
}

```

```

    fi
}

install_jq() {
    ##### INSTALL JQ TO HELP IN OUR HOOKS LATER ON #####
    echo "Installing jq"
    cd /tmp
    wget https://github.com/stedolan/jq/releases/download/jq-1.5/jq-
linux64
    chmod +x jq-linux64
    mv jq-linux64 jq
    mv jq /usr/local/bin/
}

setup_beanstalk_base() {
    echo "Creating base directories for platform."
    mkdir -p $BEANSTALK_DIR/deploy/appsource/
    mkdir -p /var/app/staging
    mkdir -p /var/app/current
    mkdir -p /var/log/nginx/healthd/
    mkdir -p /var/log/tomcat/
    apt-get -y install unzip
}

sync_platform_uploads() {
    ##### COPY THE everything in platform-uploads to / #####
    echo "Setting up platform hooks"
    rsync -ar $BUILDER_DIR/platform-uploads/ /
}

prepare_platform_base() {
    install_jq
    setup_beanstalk_base
    sync_platform_uploads
}

run_setup_scripts() {
    for entry in $( ls $BUILDER_DIR/setup-scripts/*.sh | sort ) ; do
        run_command $entry
    done
}

cleanup() {
    echo "Done all customization of packer instance. Cleaning up"
    rm -rf $BUILDER_DIR
}

set_permissions() {
    echo "Setting permissions in /opt/elasticbeanstalk"
    find /opt/elasticbeanstalk -type d -exec chmod 755 {} \; -print
    chown -R root:root /opt/elasticbeanstalk/
    echo "Setting permissions for shell scripts"
    find /opt/elasticbeanstalk/ -name "*.sh" -exec chmod 755 {} \; -
print
    echo "setting permissions done."
}

```

```
echo "Running packer builder script"
```

```
wait_for_cloudinit  
prepare_platform_base  
run_setup_scripts  
set_permissions  
cleanup
```

More samples can be found in the AWS samples repo on github:

### **aws-samples/eb-custom-platforms-samples**

Elastic Beanstalk samples for running on custom platforms. Ensure that you have the latest version of the EB CLI...

[github.com](https://github.com)

[DevOps](#)[Software Development](#)[Elastic Beanstalk](#)[AWS](#)[Migration](#)[About](#) [Help](#) [Legal](#)

Get the Medium app

