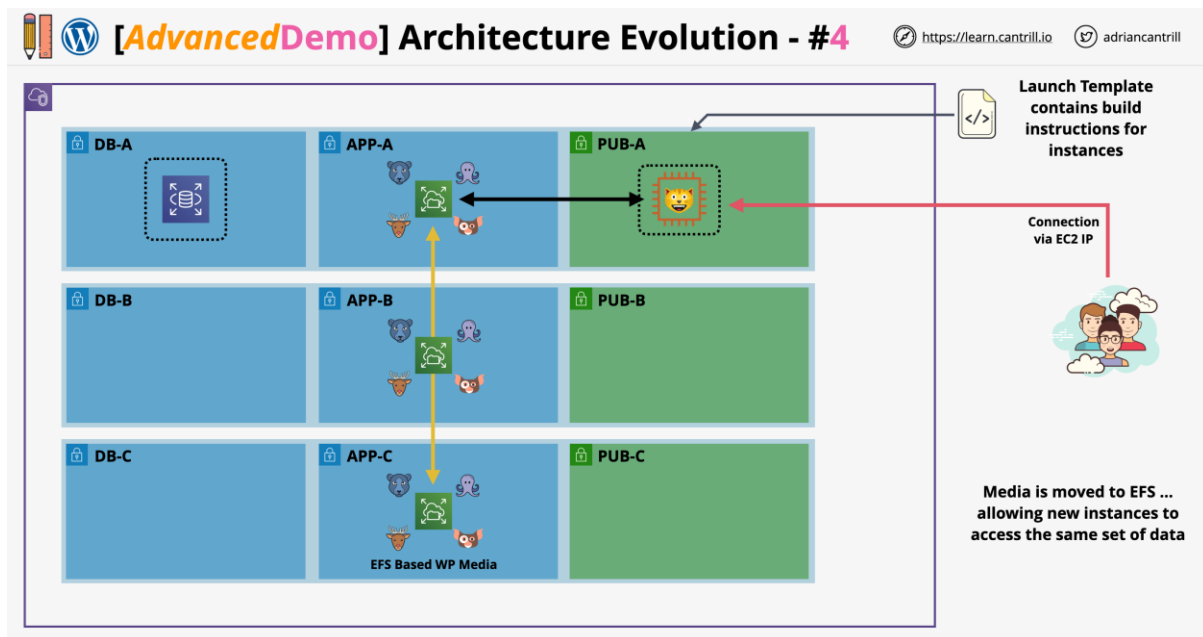


Advanced Demo - Web App - Single Server to Elastic Evolution



Welcome back, in stage 4 of this demo series you will be creating an EFS file system designed to store the wordpress locally stored media. This area stores any media for posts uploaded when creating the post as well as theme data. By storing this on a shared file system it means that the data can be used across all instances in a consistent way, and it lives on past the lifetime of the instance.

STAGE 4A - Create EFS File System

File System Settings

Move to the EFS Console <https://console.aws.amazon.com/efs/home?region=us-east-1#/get-started>

Click on Create file System

We're going to step through the full configuration options, so click on Customize

For Name type A4L-WORDPRESS-CONTENT

This is critical data so for Storage Class leave this set to Standard and ensure Enable Automatic Backups is enabled.

for LifeCycle management ...

for Transition into IA set to 30 days since last access

for Transition out of IA set to None

Untick Enable encryption of data at rest .. in production you would leave this on, but for this demo which focusses on architecture it simplifies the implementation.

Scroll down...

For throughput modes choose Bursting

Expand Additional Settings and ensure Performance Mode is set to General Purpose

Click Next

Network Settings

In this part you will be configuring the EFS Mount Targets which are the network interfaces in the VPC which your instances will connect with.

In the Virtual Private Cloud (VPC) dropdown select A4LVPC

You should see 3 rows.

Make sure us-east-1a, us-east-1b & us-east-1c are selected in each row.

In us-east-1a row, select sn-App-A in the subnet ID dropdown, and in the security groups dropdown select A4LVPC-SGEFS & remove the default security group

In us-east-1b row, select sn-App-B in the subnet ID dropdown, and in the security groups dropdown select A4LVPC-SGEFS & remove the default security group

In us-east-1c row, select sn-App-C in the subnet ID dropdown, and in the security groups dropdown select A4LVPC-SGEFS & remove the default security group

Click next

Leave all these options as default and click next

We won't be setting a file system policy so click Create

The file system will start in the Creating State and then move to Available once it does..

Click on the file system to enter it and click Network

Scroll down and all the mount points will show as creating keep hitting refresh and wait for all 3 to show as available before moving on.

Note down the fs-XXXXXXX or DNS name (either will work) once visible at the top of this screen, you will need it in the next step.

STAGE 4B - Add an fsid to parameter store

Now that the file system has been created, you need to add another parameter store value for the file system ID so that the automatically built instance(s) can load this safely.

Move to the Systems Manager console <https://console.aws.amazon.com/systems-manager/home?region=us-east-1#>

Click on Parameter Store on the left menu

Click Create Parameter

Under Name enter /A4L/Wordpress/EFSFSID Under Description enter File System ID for Wordpress Content (wp-content)

for Tier set Standard

For Type set String

for Data Type set text

for Value set the file system ID fs-XXXXXXX which you just noted down (use your own file system ID)

Click Create Parameter

STAGE 4C - Connect the file system to the EC2 instance & copy data

Open the EC2 console and go to running

instances <https://console.aws.amazon.com/ec2/v2/home?region=us-east-1#Instances:sort=desc:tag:Name>

Select the Wordpress-LT instance, right click, Connect, Select Session Manager and click Connect type sudo bash and press enter

type cd and press enter

type clear and press enter

First we need to install the amazon EFS utilities to allow the instance to connect to EFS. EFS is based on NFS which is standard but the EFS tooling makes things easier.

```
sudo dnf -y install amazon-efs-utils
```

next you need to migrate the existing media content from wp-content into EFS, and this is a multi step process.

First, copy the content to a temporary location and make a new empty folder.

```
cd /var/www/html
```

```
sudo mv wp-content/ /tmp
```

```
sudo mkdir wp-content
```

then get the efs file system ID from parameter store

```
EFSFSID=$(aws ssm get-parameters --region us-east-1 --names /A4L/Wordpress/EFSFSID --query Parameters[0].Value)
```

```
EFSFSID=`echo $EFSFSID | sed -e 's/^"/' -e 's/"$/'`
```

Next .. add a line to /etc/fstab to configure the EFS file system to mount as /var/www/html/wp-content/

```
echo -e "$EFSFSID:/var/www/html/wp-content efs _netdev,tls,iam 0 0" >> /etc/fstab
```

```
mount -a -t efs defaults
```

now we need to copy the origin content data back in and fix permissions

```
mv /tmp/wp-content/* /var/www/html/wp-content/
```

```
chown -R ec2-user:apache /var/www/
```

STAGE 4D - Test that the wordpress app can load the media

run the following command to reboot the EC2 wordpress instance

```
reboot
```

Once it restarts, ensure that you can still load the wordpress blog which is now loading the media from EFS.

STAGE 4E - Update the launch template with the config to automate the EFS part

Next you will update the launch template so that it automatically mounts the EFS file system during its provisioning process. This means that in the next stage, when you add autoscaling, all instances will have access to the same media store ...allowing the platform to scale.

Go to the EC2 console <https://console.aws.amazon.com/ec2/v2/home?region=us-east-1#Home:>

Click Launch Templates

Check the box next to the Wordpress launch template, click Actions and click Modify Template (Create New Version)

for Template version description enter App only, uses EFS filesystem defined in /A4L/Wordpress/EFSFSID

Scroll to the bottom and expand Advanced Details

Scroll to the bottom and find User Data expand the entry box as much as possible.

After #!/bin/bash -xe position cursor at the end & press enter twice to add new lines paste in this

```
EFSFSID=$(aws ssm get-parameters --region us-east-1 --names /A4L/Wordpress/EFSFSID --query Parameters[0].Value)
```

```
EFSFSID=`echo $EFSFSID | sed -e 's/^"/' -e 's/"$/'`
```

Find the line which says dnf install wget php-mysqldb httpd php-fpm php-mysqli mariadb105-server php-json php php-devel stress -y after stress add a space and paste in amazon-efs-utils it should now look like dnf install wget php-mysqldb httpd php-fpm php-mysqli mariadb105-server php-json php php-devel stress amazon-efs-utils -y

locate systemctl start httpd position cursor at the end & press enter twice to add new lines

paste in the following

```
mkdir -p /var/www/html/wp-content
```

```
chown -R ec2-user:apache /var/www/
```

```
echo -e "$EFSFSID:/ /var/www/html/wp-content efs _netdev,tls,iam 0 0" >> /etc/fstab
```

```
mount -a -t efs defaults
```

Scroll down and click Create template version

Click View Launch Template

Select the template again (dont click) Click Actions and select Set Default Version

Under Template version select 3

Click Set as default version

STAGE 4 - FINISH

This configuration has several limitations :-

- ~~The application and database are built manually, taking time and not allowing automation~~ FIXED
- ~~^^ it was slow and annoying ... that was the intention.~~ FIXED
- ~~The database and application are on the same instance, neither can scale without the other~~ FIXED
- ~~The database of the application is on an instance, scaling IN/OUT risks this media~~ FIXED
- ~~The application media and UI store is local to an instance, scaling IN/OUT risks this media~~ FIXED
- Customer Connections are to an instance directly ... no health checks/auto healing
- The IP of the instance is hardcoded into the database

You can now move onto STAGE 5