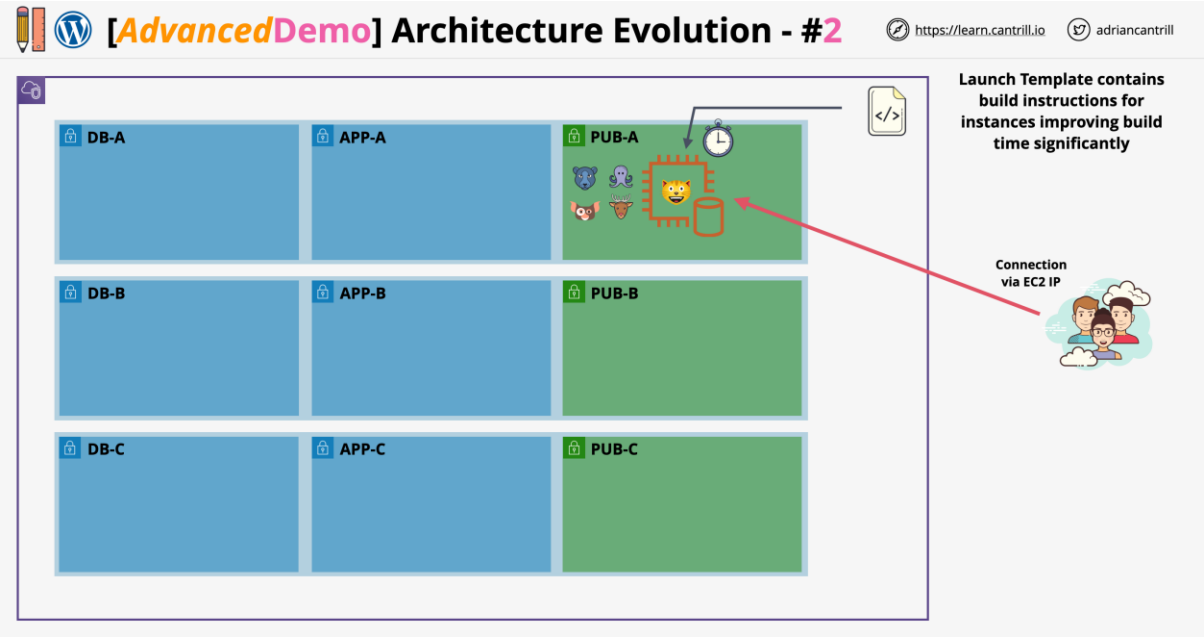
Advanced Demo - Web App - Single Server to Elastic Evolution



In stage 2 of this advanced demo lesson you are going to create a launch template which can automate the build of WordPress.
The architecture will still use the single instance for both the WordPress application and database, the only change will be an automatic build rather than manual.
Any level of automation/self-healing or scaling architecture will need a bootstrapped or AMI-baked build to function effectively.

STAGE 1 TIDYUP

Right click on the manual instance you created in the previous step called Wordpress-Manual and select Terminate Instance and confirm that termination.

STAGE 2A - Create the Launch Template

Open the EC2 console https://console.aws.amazon.com/ec2/v2/home?region=us-east-1#Instances:sort=desc:tag:Name
Click Launch Templates under Instances on the left menu
Click Create Launch Template
Under Launch Template Name enter Wordpress
Under Template version description enter Single server DB and App
Check the Provide guidance to help me set up a template that I can use with EC2 Auto Scaling box

Under Application and OS Images (Amazon Machine Image) click Quick Start
Click Amazon Linux
in the Amazon Machine Image dropdown, locate Amazon Linux 2023 AMI and set the Architecture to 64-bit (x86)
Under Instance Type select whichever instance is free tier eligible from either t3.micro and t2.micro
Under Key pair (login) select Don't include in launch template
Under networking Settings select existing security group and choose A4LVPC-SGWordpress Leave storage volumes unchanged
Leave Resource Tags Unchanged

Expand Advanced Details Under IAM instance profile select A4LVPC-WordpressInstanceProfile there will be some random at the end, thats ok!
Under Credit specification select Standard

STAGE 2B - Add Userdata

At this point we need to add the configuration which will build the instance Enter the user data below into the User Data box

```
#!/bin/bash -xe


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

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


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

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


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

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


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

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


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

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


dnf -y update


dnf install wget php-mysqlnd httpd php-fpm php-mysqli mariadb105-server php-json php php-devel stress -y


systemctl enable httpd
```

```
systemctl enable mariadb
systemctl start httpd
systemctl start mariadb

mysqladmin -u root password $DBRootPassword

wget http://wordpress.org/latest.tar.gz -P /var/www/html
cd /var/www/html
tar -zxvf latest.tar.gz
cp -rvf wordpress/* .
rm -R wordpress
rm latest.tar.gz

sudo cp ./wp-config-sample.php ./wp-config.php
sed -i "s/'database_name_here'/'$DBName'/g" wp-config.php
sed -i "s/'username_here'/'$DBUser'/g" wp-config.php
sed -i "s/'password_here'/'$DBPassword'/g" wp-config.php
sed -i "s/'localhost'/'$DBEndpoint'/g" wp-config.php

usermod -a -G apache ec2-user
chown -R ec2-user:apache /var/www
chmod 2775 /var/www
find /var/www -type d -exec chmod 2775 {} \;
find /var/www -type f -exec chmod 0664 {} \;

echo "CREATE DATABASE $DBName;" >> /tmp/db.setup
echo "CREATE USER '$DBUser'@'localhost' IDENTIFIED BY '$DBPassword';" >> /tmp/db.setup
echo "GRANT ALL ON $DBName.* TO '$DBUser'@'localhost';" >> /tmp/db.setup
echo "FLUSH PRIVILEGES;" >> /tmp/db.setup
mysql -u root --password=$DBRootPassword < /tmp/db.setup
rm /tmp/db.setup
```

Ensure to leave a blank line at the end
Click Create Launch Template
Click View launch templates

STAGE 2C - Launch an instance using it

Select the launch template in the list ... it should be called Wordpress
Click Actions and Launch instance from template Under Key Pair(login click the dropdown and select Proceed without a key pair(Not Recommended)
Scroll down to Network settings and under Subnet select sn-pub-A
Scroll to Resource Tags click Add tag Set Key to Name and Value to Wordpress-LT Scroll to the bottom and click Launch Instance
Click the instance id in the Success box

STAGE 2D - Test

you will need to wait until the instance is running with 2/2 status checks before contuining

Open the EC2 console https://console.aws.amazon.com/ec2/v2/home?region=us-east-1#Instances:sort=desc:tag:Name
Select the Wordpress-LT instance
copy the IPv4 Public IP into your clipboard, don't click the link to open, this will use https and we want http, just copy the IP.
Open that IP in a new tab
You should see the WordPress welcome page

Perform Initial COnfiguration and make a post

in Site Title enter Catagram
in Username enter admin in Password enter 4n1m4l54L1f3
in Your Email enter your email address
Click Install WordPress Click Log In
In Username or Email Address enter admin
in Password enter the previously noted down strong password Click Log In

Click Posts in the menu on the left
Select Hello World! Click Bulk Actions and select Move to Trash Click Apply

Click Add New
If you see any popups close them down
For title The Best Animal(s)!
Click the + under the title, select Gallery Click Upload
Select some animal pictures.... if you don't have any use google images to download some
Upload them
Click Publish
Click Publish Click view Post

This is your working, auto built WordPress instance ** don't terminate the instance this time - we're going to migrate the database in stage 3**

STAGE 2 - 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
- The database of the application is on an instance, scaling IN/OUT risks this media
- The application media and UI store is local to an instance, scaling IN/OUT risks this media
- 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 STAGE3