ENT303

Best practices for containerizing legacy applications

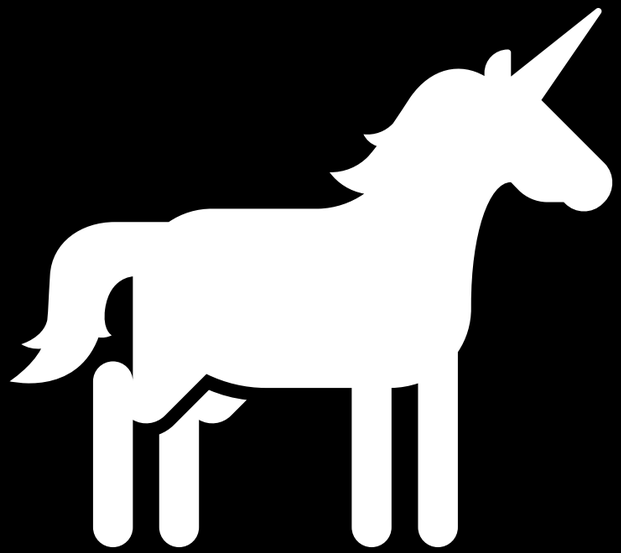
Diego Dalmolin

Sr. Solutions Architect, Migrations
AWS

AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

The legacy application



“Let’s containerize this legacy application”



AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Container benefits



Agility

Develop and deploy faster to achieve business goals



Portable

Portable, scalable, and isolated application deployment



Efficiency

Reduce IT operational overhead, and achieve optimized computing infrastructure

Challenges



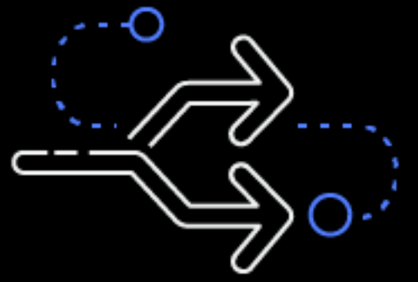
Technology

Legacy OS, platform, applications, and architecture



Staff

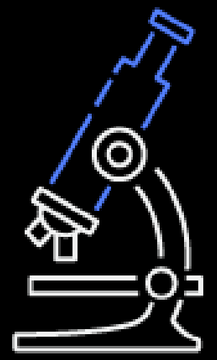
Lack of expertise, busy day-to-day, lower priority



Operations

Change in deployment, development and support model

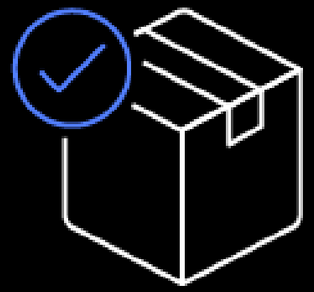
Steps



Discover



Prepare



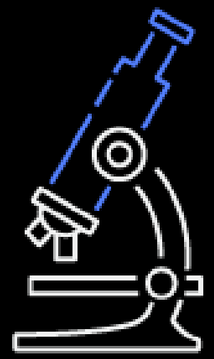
Build



Launch

AWS App2Container

https://aws.amazon.com/app2container/

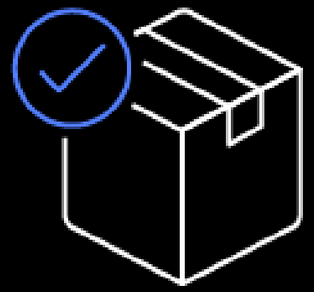


Discover

Java
.NET



Prepare



Build

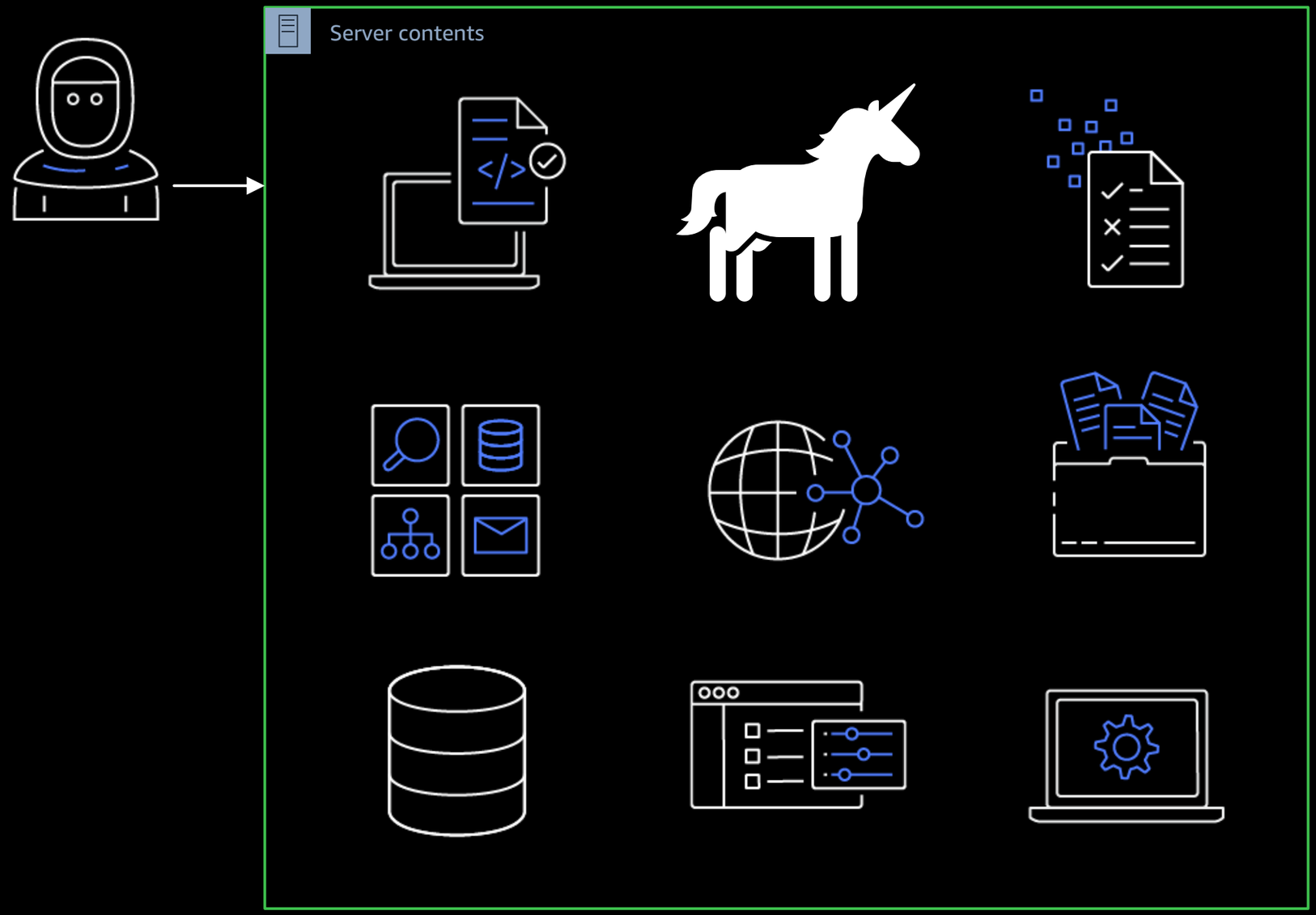
Docker



Launch

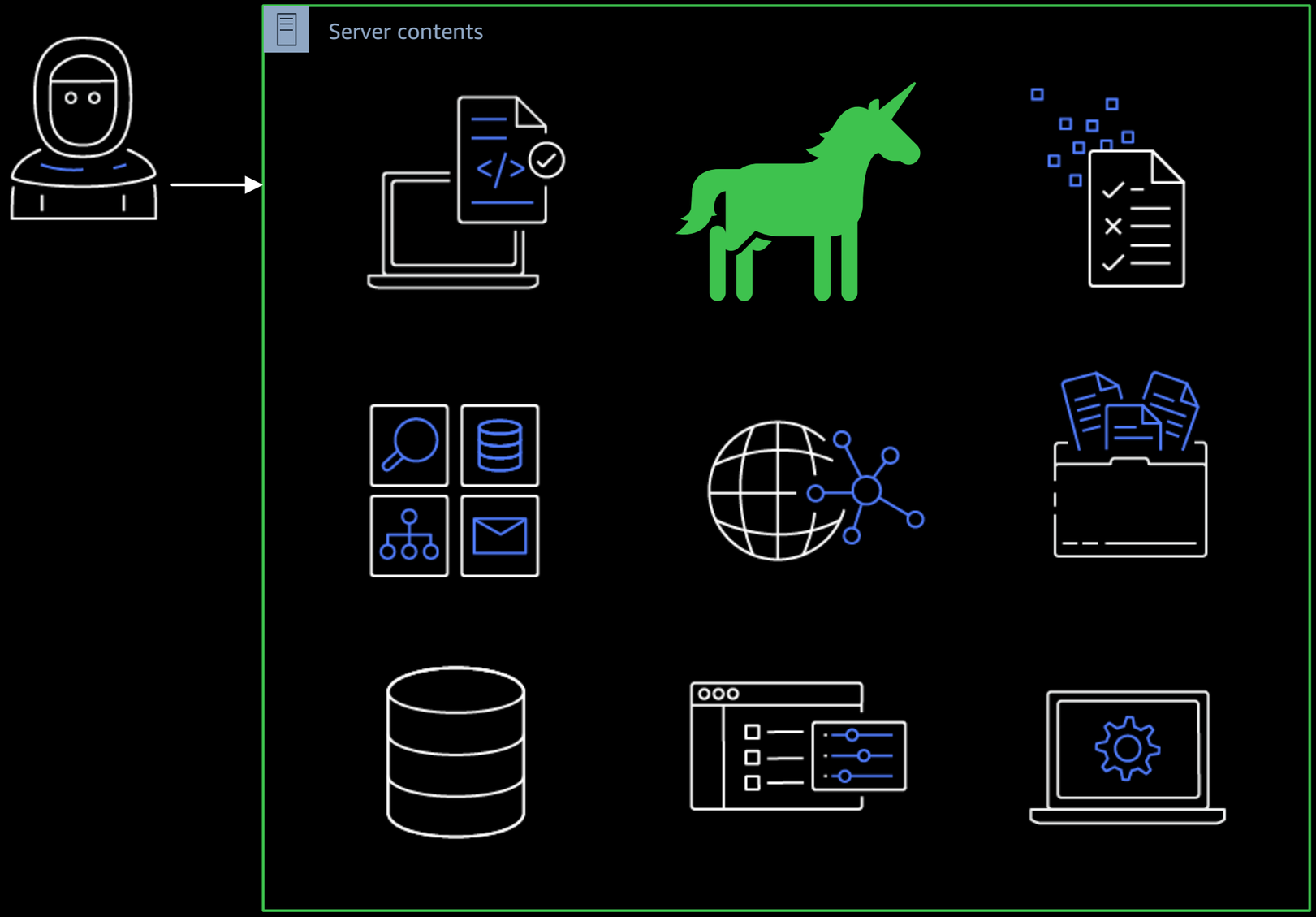
Amazon ECS
AWS Fargate
Amazon EKS
AWS CodePipeline

Discover



Discover

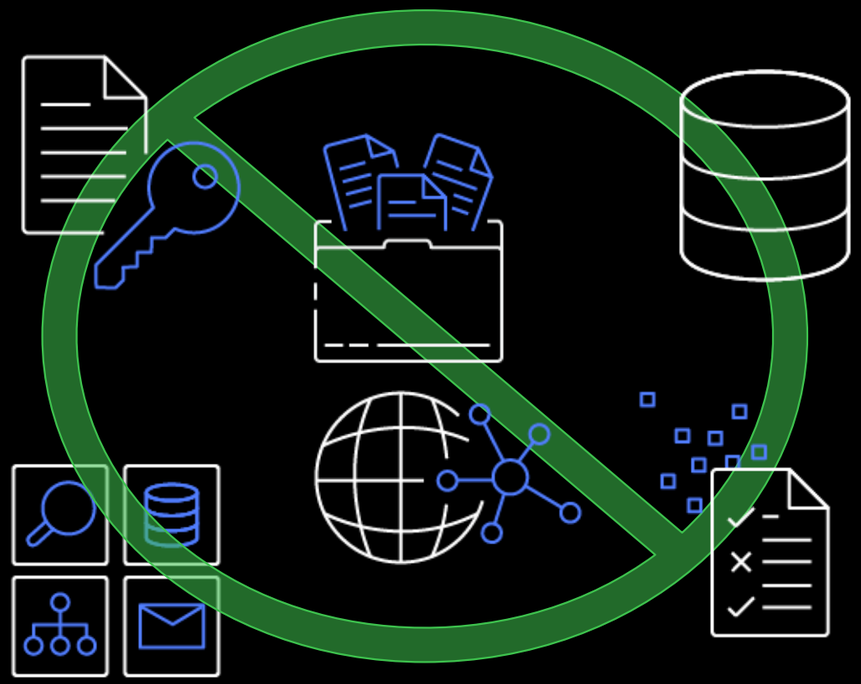
- Network
- Protocols
- Security
- Platform
  - Binary
  - Configuration
  - Logs
- Application
  - Binary
  - Source code
  - Configuration
  - Connection strings
  - Data
  - Logs
  - Sessions
  - Other dependencies
- Scheduled tasks
- Operating system



	Easy	Moderate	Difficult
Binary	Single process	Multiple process	Self-modifying
Language	Modern programming languages	X86 precompiled binaries	Non-X86
Configuration	Single file	Several files	Everywhere
Data	Single folder	Multiple folders / partitions	Everywhere
Secrets	Static files	Network	Dynamic certificates
Sessions	Stateless	Stateful	Cluster dependent
Interface	Web, backend processing	Services, background process, scheduled jobs, agents, daemons	Desktop client
Network	HTTP / HTTPS	TCP, UDP	IPSEC, highly isolated
Installation	Packages, source	Unattended installer	Interactive installer
Licensing	Self / open source	Proprietary / no source	Restrictive

Prepare

- Portable
  - NO passwords / secrets
  - NO connection strings
  - NO other dependencies (if possible)
- Ephemeral
  - NO data
  - NO database data
  - NO logs
  - NO performance statistics

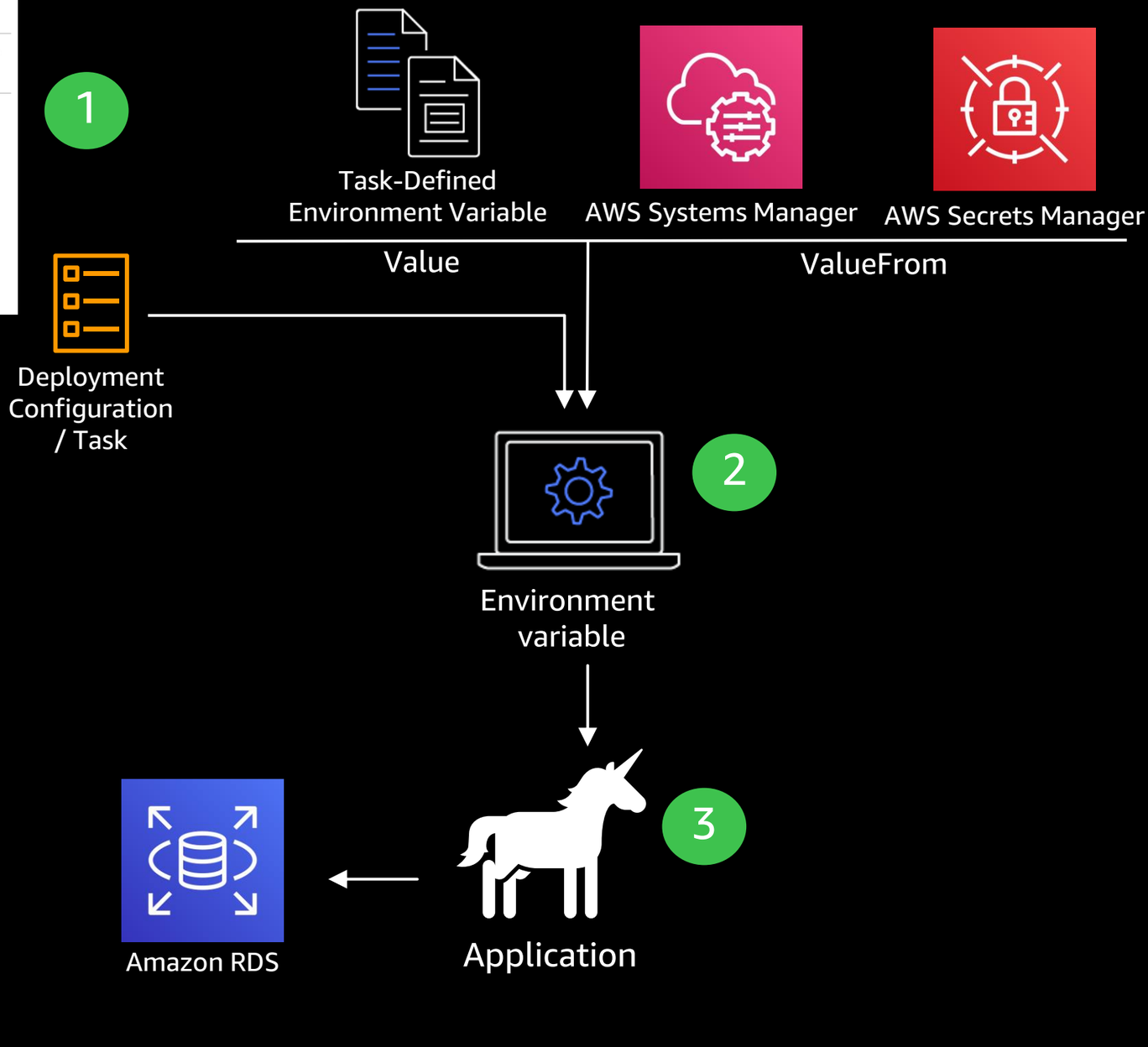


Prepare – portable

Environment variables

You may also designate AWS Systems Manager Parameter Store keys or ARNs using the 'valueFrom' field. ECS will inject the value into containers at run-time.

Key		
RDS_DB_NAME	Value	mydb
RDS_USERNAME	*Value*	myuser
RDS_PASSWORD	ValueFrom	arn:aws:secretsmanager:us-west-2:857
RDS_HOSTNAME	Value	mydb-instance-1.us-west-2.rds.amazo



Prepare – portable – source code



```
getLocalConnection() {
...
String dbName = "mydb";
String userName = "mydbuser";
String password ="mydbpassword";
String hostname = "192.168.0.100";
String port = "5432";
   String jdbcUrl = "jdbc:postgresql://"
   + hostname + ":" + port + "/" + dbName
   + "?user=" + userName
   + "&password=" + password;
...
}
```



```
getLocalConnection() {
...
String dbName = System.getenv("DB_NAME");
String userName = System.getenv("DB_USERNAME");
String password = System.getenv("DB_PASSWORD");
String hostname = System.getenv("DB_HOSTNAME");
String port = System.getenv("DB_PORT");
   String jdbcUrl = "jdbc:postgresql://"
   + hostname + ":" + port + "/" + dbName
   + "?user=" + userName
   + "&password=" + password;
...
}
```

Prepare – portable – configuration file



myapp.conf

```
// ** MySQL settings ** //
define('DB_NAME','mydb');
define('DB_USER','mydbuser');
define('DB_PASSWORD','mydbpassword');
define('DB_HOST','192.168.0.100');
```

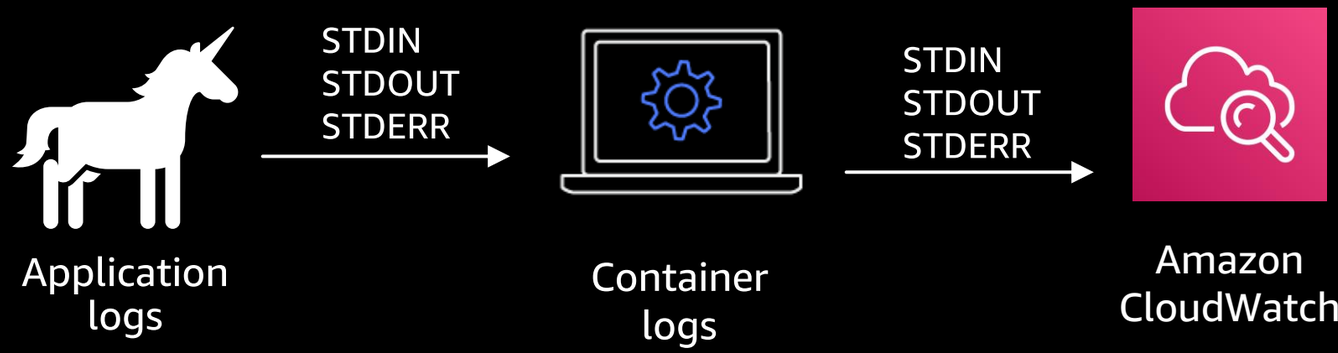


myapp.conf

```
// ** MySQL settings ** //
define('DB_NAME','ENV_DB_NAME');
define('DB_USER','ENV_DB_USERNAME');
define('DB_PASSWORD','ENV_DB_PASSWORD');
define('DB_HOST','ENV_DB_HOSTNAME');
```

```
#set database details with sed find and replace
sed -i "s/ENV_DB_NAME/$DB_NAME/" myapp.conf
sed -i "s/ENV_DB_USERNAME/$DB_USERNAME/" myapp.conf
sed -i "s/ENV_DB_PASSWORD/$DB_PASSWORD/" myapp.conf
sed -i "s/ENV_DB_HOSTNAME/$DB_HOSTNAME/" myapp.conf
#start myapp
```

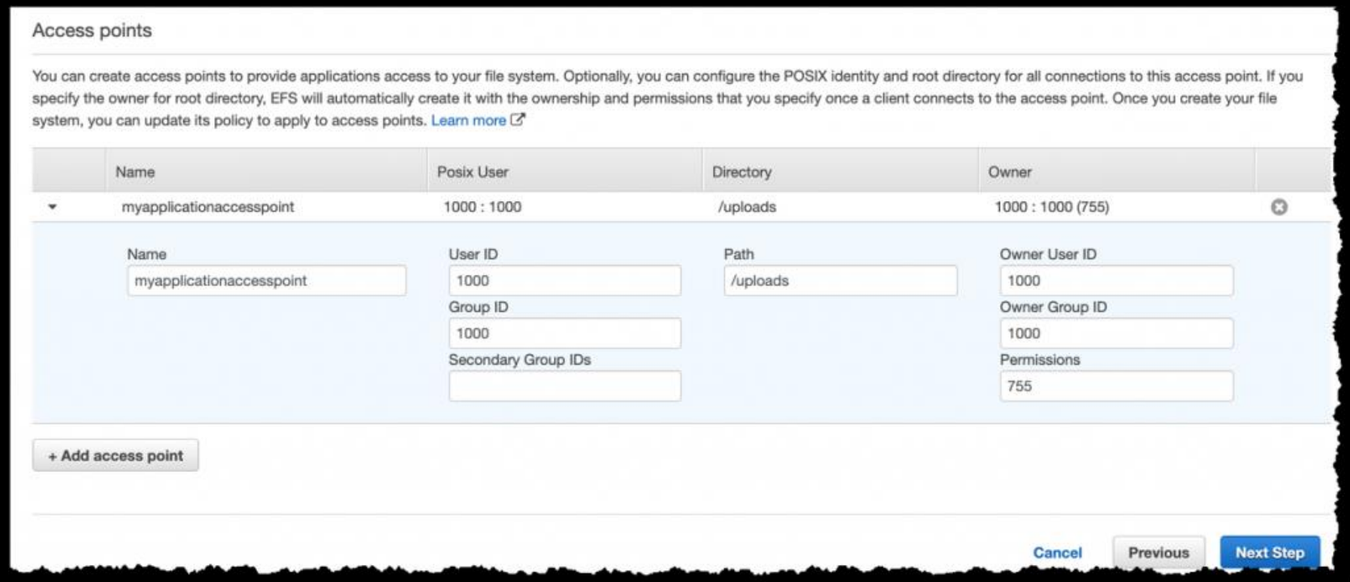
Prepare – ephemeral – logs

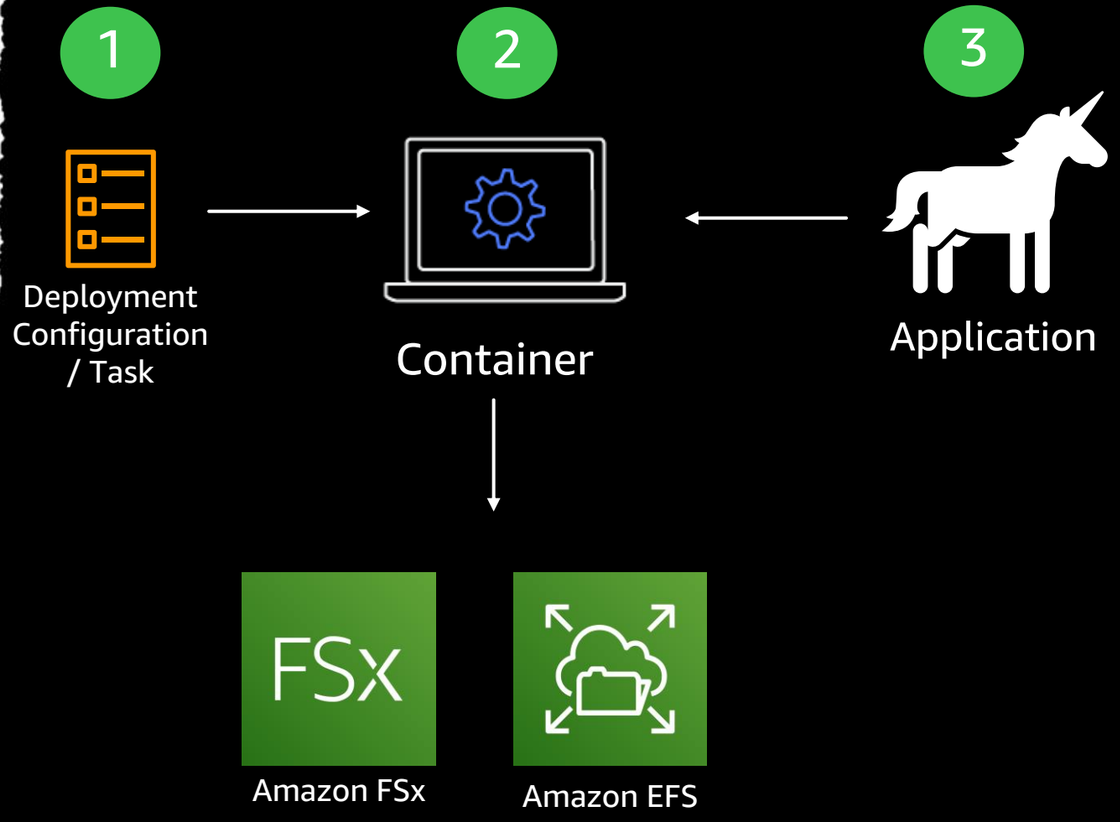


Filter events | all 30s 5m 1h 6h 1d 1w custom

Time (UTC -07:00)	Message
2016-09-09	
	No older events found at the moment. Retry.
12:56:47	WordPress not found in /var/www/html - copying now...
12:56:47	Complete! WordPress has been successfully copied to /var/www/html
12:56:49	AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 1…
12:56:49	AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 1…
12:56:49	[Fri Sep 09 19:56:49.059245 2016] [mpm_prefork:notice] [pid 1] AH00163: Apache/2.4.10 (Debian…
12:56:49	[Fri Sep 09 19:56:49.059273 2016] [core:notice] [pid 1] AH00094: Command line: 'apache2 -D FO…
13:06:55	52.90.111.181 - - [09/Sep/2016:20:06:55 +0000] "GET /api/json HTTP/1.1" 302 377 "-" "python-r…
13:06:56	52.90.111.181 - - [09/Sep/2016:20:06:55 +0000] "GET /wp-admin/install.php HTTP/1.1" 200 344…
13:06:56	52.90.111.181 - - [09/Sep/2016:20:06:56 +0000] "\x16\x03\x01" 400 0 "-" "-"
13:06:57	54.210.246.190 - - [09/Sep/2016:20:06:57 +0000] "GET http://example.com/ HTTP/1.1" 302 376 …

Prepare – ephemeral – data





Build and launch

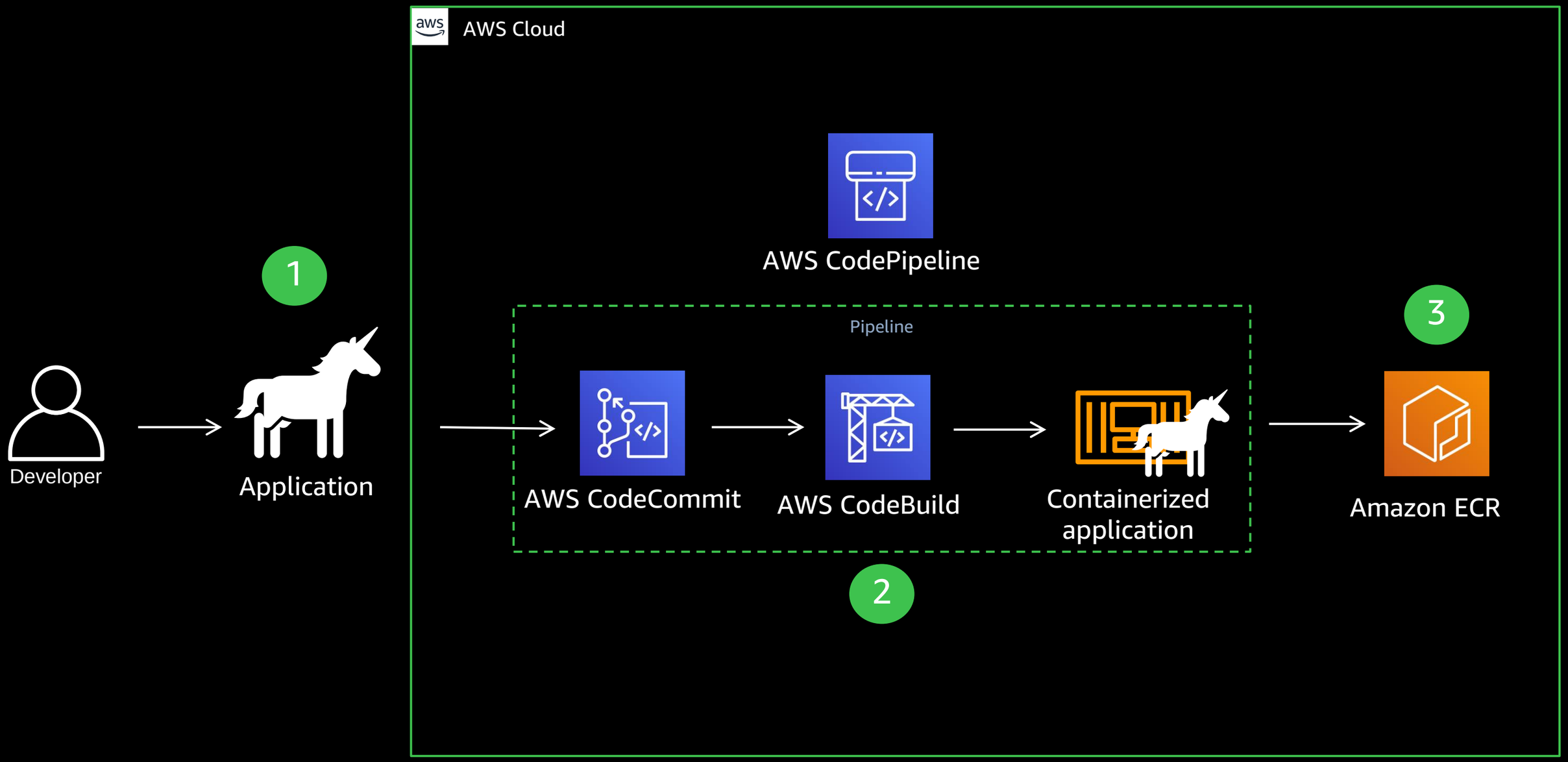
Dockerfile

```
# lightweight base image.
FROM mybaseimage:latest
...
# redirect application logs
RUN ln -sf /dev/stdout /myapp/myapp.log
# Fix my configuration file.
CMD [ "fixconf.sh" ]
# Run the specified command.
CMD [ "catalina.sh", "run" ]
```

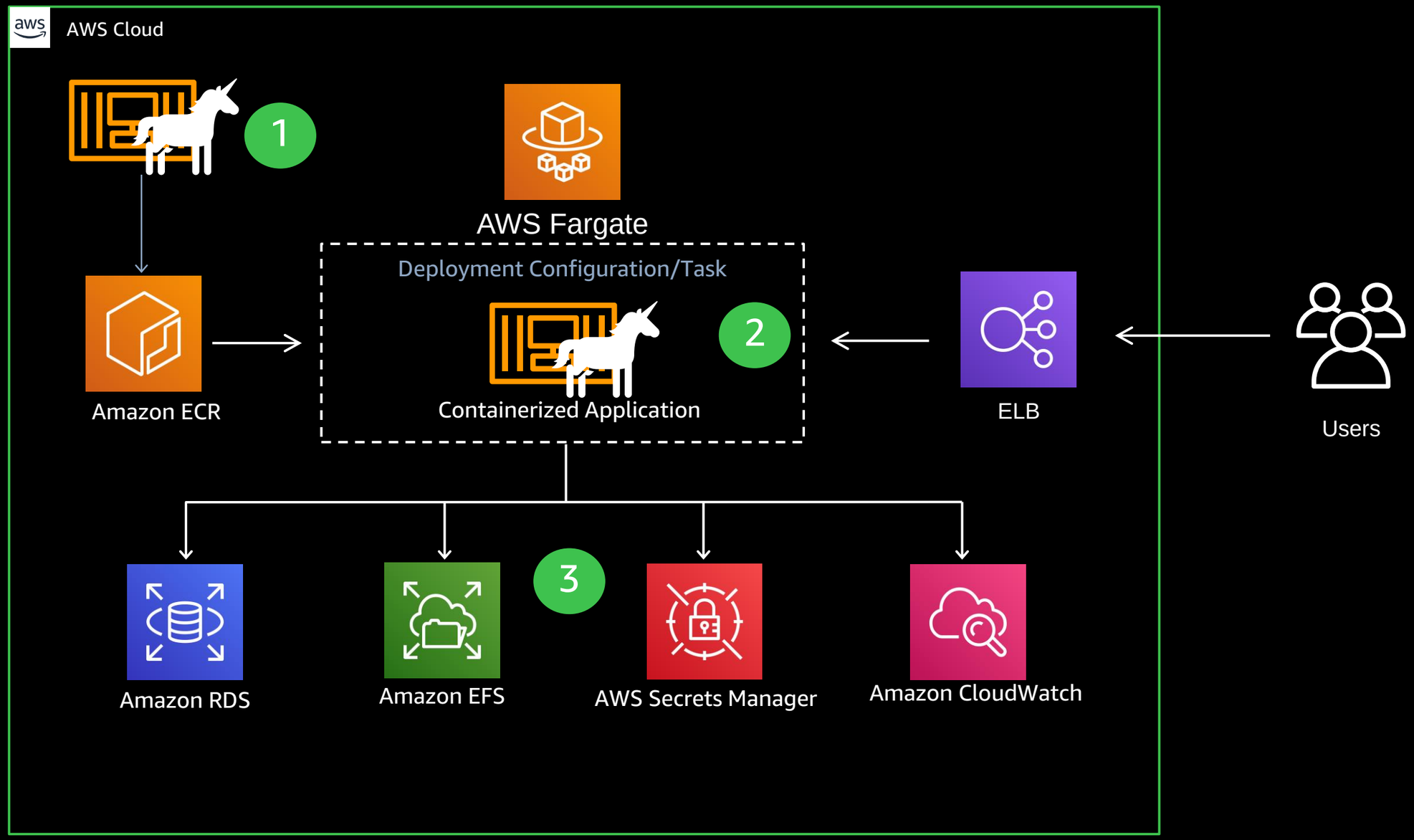
ECS Task

```
...
"image": "myapp:latest",
"logConfiguration": {
   "logDriver": "awslogs"},
...
"environment" : [
   {"name":"DB_NAME","value":"mydb"},
   {"name":"DB_USER","value":"mydbuser"}],
"secrets": [{
   "name": "DB_PASSWORD",
   "valueFrom":“ARN:…secretsmanager…“}],
"mountPoints": [{
   "sourceVolume":"myEfsVolume",
   "containerPath":"/uploads"}]
...
```

Build


AWS Cloud
AWS CodePipeline
1
Developer
Application
Pipeline
AWS CodeCommit
AWS CodeBuild
Containerized application
2
3
Amazon ECR

Launch


AWS Cloud
1
AWS Fargate
Deployment Configuration/Task
Amazon ECR
Containerized Application
2
ELB
Users
3
Amazon RDS
Amazon EFS
AWS Secrets Manager
Amazon CloudWatch

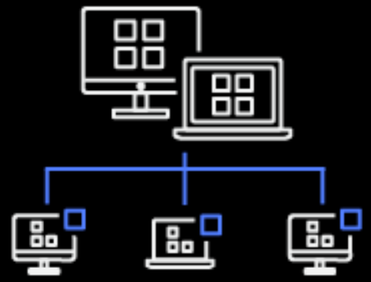
Next steps



AWS



Experiment



Reuse



Break the monolith

Resources

Cloud Migration Homepage: https://aws.amazon.com/cloud-migration/

Additional Migration Resources,

AWS Prescriptive Guidance,

AWS Enterprise Blog Collection:
https://aws.amazon.com/cloud-migration/resources/

Migration Immersion Day: https://migration-immersionday.workshop.aws/en

All AWS workshops available here: https://www.workshops.aws/



© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Thank you!



AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Please complete the session survey

AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.