**AWS ECS**

**Ref:**

ECS Overview:

<https://ecsworkshop.com/introduction/ecs_basics/>

ECS components : <https://docs.aws.amazon.com/AmazonECS/latest/developerguide/welcome-features.html>

ECS vs K8s:

<https://platform9.com/blog/kubernetes-vs-ecs/>

ECS cli:

<https://docs.aws.amazon.com/cli/latest/reference/ecs/index.html>

Harshicorp Waypoint:

<https://www.waypointproject.io/docs/intro>

ECS using waypoint:

<https://learn.hashicorp.com/tutorials/waypoint/aws-ecs>

CoPilot-cli:

<https://docs.aws.amazon.com/AmazonECS/latest/developerguide/getting-started-aws-copilot-cli.html>

Start task at container instance launch time

<https://docs.aws.amazon.com/AmazonECS/latest/developerguide/start_task_at_launch.html>

Terraform-ECS

<https://github.com/tmknom/terraform-aws-ecs-fargate>

Creating a cluster with an EC2 task using the AWS CLI

<https://docs.aws.amazon.com/AmazonECS/latest/developerguide/ECS_AWSCLI_EC2.html>

**AWS CodePipeline**

Lambda functions can be used in CI/CD pipelines to do the following tasks:

• Roll out changes to your environment by applying or updating an AWS CloudFormation template.

• Create resources on demand in one stage of a pipeline using AWS CloudFormation and delete them in another stage.

• Deploy application versions with zero downtime in AWS Elastic Beanstalk with a Lambda function that swaps Canonical Name record (CNAME) values.

• Deploy to Amazon EC2 Container Service (ECS) Docker instances.

• Back up resources before building or deploying by creating an AMI snapshot.

• Add integration with third-party products to your pipeline, such as posting messages to an Internet Relay Chate (IRC) client.

Ref:

[Lambda Function for AMI backup](https://www.bluepiit.com/blog/how-to-automate-ami-backups-cleanups-using-aws-lambda-serverless-with-ec2-tags/)

CodeDeploy deployment on EC2:

<https://seanjziegler.com/deploying-code-from-github-to-aws-ec2-with-codepipeline/>

CodeBuild VPC support:

<https://docs.aws.amazon.com/codebuild/latest/userguide/vpc-support.html>

CodeDeploy Components:

<https://docs.aws.amazon.com/codedeploy/latest/userguide/primary-components.html>

AppSpec File Structure:

<https://docs.aws.amazon.com/codedeploy/latest/userguide/reference-appspec-file-structure.html>

CodePipeline with VPC:

<https://docs.aws.amazon.com/codepipeline/latest/userguide/vpc-support.html> (Uses AWS PrivateLink / VPC endpoints)

AWS SAM

<https://docs.aws.amazon.com/serverlessrepo/latest/devguide/what-is-serverlessrepo.html>

AWS SAM hello world app

<https://docs.aws.amazon.com/serverless-application-model/latest/developerguide/serverless-getting-started-hello-world.html>

SAM resource:

<https://docs.aws.amazon.com/serverless-application-model/latest/developerguide/sam-resource-function.html>

AWS Lambda and SAM tutorial

<https://docs.aws.amazon.com/lambda/latest/dg/applications-tutorial.html>

AWS lambda with Api Gateway

<https://docs.aws.amazon.com/lambda/latest/dg/services-apigateway-tutorial.html>

CodeCommit Clone in local:

git config --global credential.helper "!aws codecommit credential-helper $@"

git config --global credential.UseHttpPath true

**Host Networking in Docker**

The container does not get its own IP-address allocated. For instance, if you run a container which binds to port 80 and you use host networking, the container’s application is available on port 80 on the host’s IP address.

Note: Given that the container does not have its own IP-address when using host mode networking, port-mapping does not take effect, and the -p, --publish, -P, and --publish-all option are ignored

**GitHub**

Terraform AWS Resources:

<https://github.com/terraform-aws-modules>

AWS github repo:

<https://github.com/orgs/aws-samples/repositories>

AWS ECS repo:

<https://github.com/orgs/aws-containers/repositories>

Terraform AWS Provider:

<https://github.com/hashicorp/terraform-provider-aws>

Amazon-ecs-fullstack-app-terraform

<https://github.com/aws-samples/amazon-ecs-fullstack-app-terraform>

**Github and Gitlab sync**

<https://dev.to/brunorobert/github-and-gitlab-sync-44mn>

# **Duplicating a repository**

<https://docs.github.com/en/repositories/creating-and-managing-repositories/duplicating-a-repository>

Nodejs serverless framework demo:

<https://github.com/sd031/serverless-framework-ci-cd-aws/blob/master/buildspec.yml>