AWS DevOps

Olga Khorkova

Developer Open Space

17. Dezember 2021

Inhalte

Paxis

- 5 mini projects (3 of them are individual)
- Resources:
  `https://github.com/pr-olga/ws-aws-devops-2021`

Agenda

- Start 9 a.m.
- Lunch 12 - 1 p.m.
- End 4 p.m.
- Short breaks (10-15 min) after each block (the first one is around 10:30 a.m.)

Purpose of the day

By the end of the day, (i) you have got a general overview of AWS, (ii) have an idea how to migrate/create a new project into/in AWS, and (iii) ask yourself how you could live without it!?

About me

- Olga Khorkova
- Fullstack dev (JS/PHP) + DevOps
- @ horizn-studios.com
- Twitter @prolga_

Intro

Why Cloud?

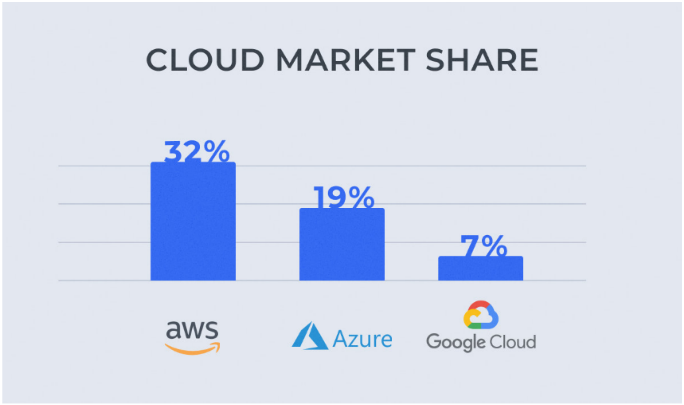
- it saves time, save the costs, with minimal amount of people you can create great products
- no (physical) servers to take care of
- continuously scale
- run on demand
- pay if code runs

Story/Market Comparison

Start 2000 -> market leadership 2016

Around the same time-frame, Amazon was frustrated with the speed of its software engineering, and sought to implement various recommendations put forth by Matt Round, an engineering leader at the time, including ***maximization of autonomy for engineering teams, adoption of REST, standardization of infrastructure, removal of gate-keeping decision-makers (bureaucracy), and continuous deployment****. He also called for increasing the percentage of the time engineers spent building the software rather than doing other tasks.[24] Amazon created a shared IT platform so its engineering organizations which were spending 70% of their time on undifferentiated heavy-lifting such as IT and infrastructure problems could focus on customer-facing innovation instead.[25] Besides, in dealing with unusual peak traffic patterns especially during the holiday season, migrating services to commodity Linux hardware, and reliance on open source software already had Amazon's Infrastructure team, led by Tom Killalea,[26] Amazon's first CISO,[27] run their data centers and associated services in a fast, reliable, cheap way.[26]*
source: wikipedia

Story/Market Comparison



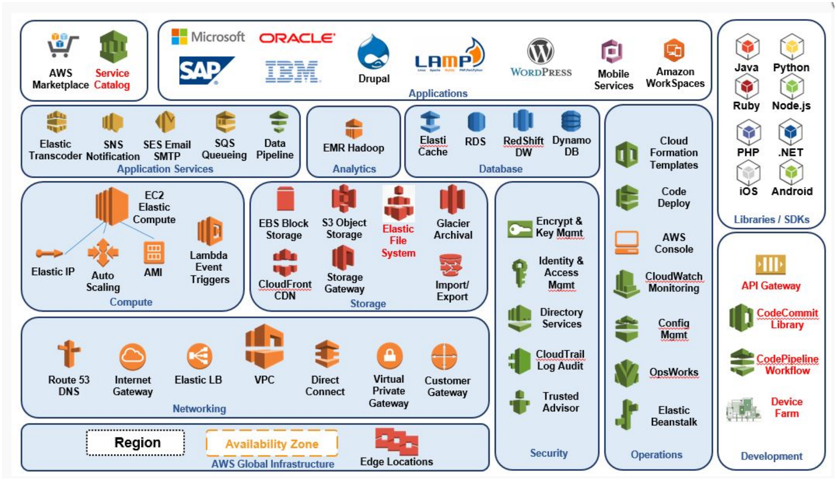
Global Infrastructure

AWS is a leader: `https://aws.amazon.com/about-aws/global-infrastructure/`
`https://aws.amazon.com/about-aws/global-infrastructure/regions_az/`

- Region: Geographical Region
  - dots on the maps
  - it has availability zoness (AZ)
  - they are independent
- Edge Location: Used to cache files
  - mini data center
  - for cahcing files only used closer to a users location
- Availability Zone: physical data center within a specific region
  - there are multiple data centers in a given region
  - isolated location within a geographic region

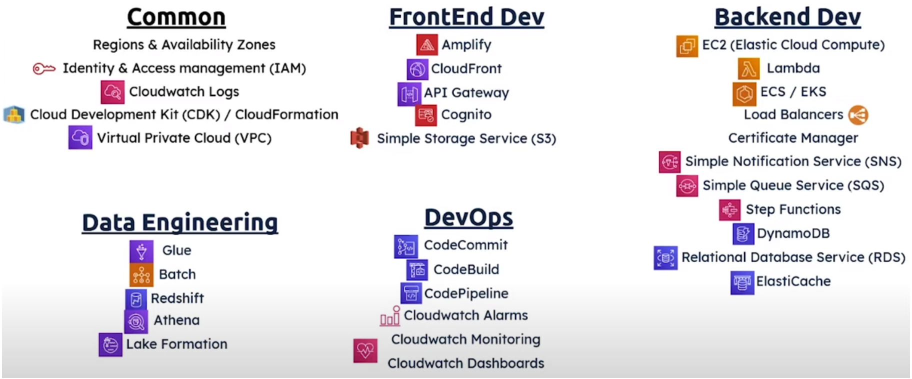
Services: Classification 1

As of 2021, AWS comprises over 200 products and services



Services: Classification 2

source: https://www.youtube.com/watch?v=N8lcedBPmE8



Common

Regions & Availability Zones
Identity & Access management (IAM)
Cloudwatch Logs
Cloud Development Kit (CDK) / CloudFormation
Virtual Private Cloud (VPC)

FrontEnd Dev

Amplify
CloudFront
API Gateway
Cognito
Simple Storage Service (S3)

Backend Dev

EC2 (Elastic Cloud Compute)
Lambda
ECS / EKS
Load Balancers
Certificate Manager
Simple Notification Service (SNS)
Simple Queue Service (SQS)
Step Functions
DynamoDB
Relational Database Service (RDS)
ElastiCache

Data Engineering

Glue
Batch
Redshift
Athena
Lake Formation

DevOps

CodeCommit
CodeBuild
CodePipeline
Cloudwatch Alarms
Cloudwatch Monitoring
Cloudwatch Dashboards

Shared Resposibily Model

- `https://aws.amazon.com/compliance/shared-responsibility-model/`
- AWS is responsible for security OF the cloud, we are responsible for security IN the cloud.

Pricing

- Price Calculator `https://calculator.aws/`
- Free tiers `https://aws.amazon.com/free/`

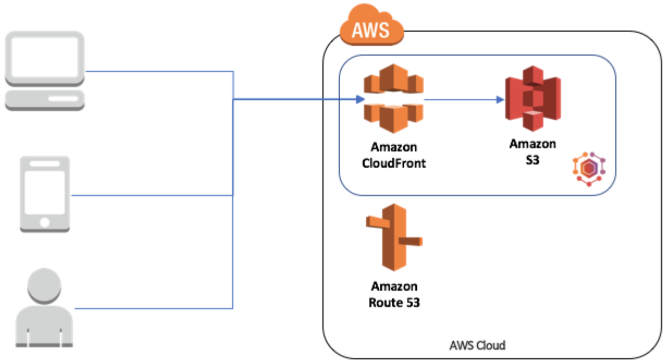
Architectural Solutions

Architectural Solutions: each architectural challenge can have many solutions inside of AWS!

Project 0

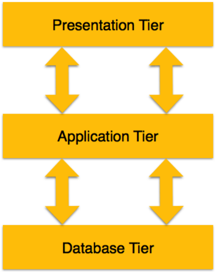
what resources/configs do we need if we create an online magazine?

Static Page



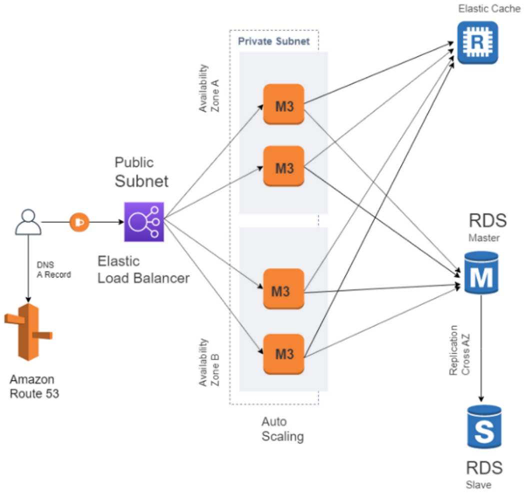
3-tiers architecture 1

Software Entwicklung = MVC
source
`https://medium.com/@codewithniraj/the-typical-3-tier-architecture-of-web-application-in-aws-f2f9d662fdfe`



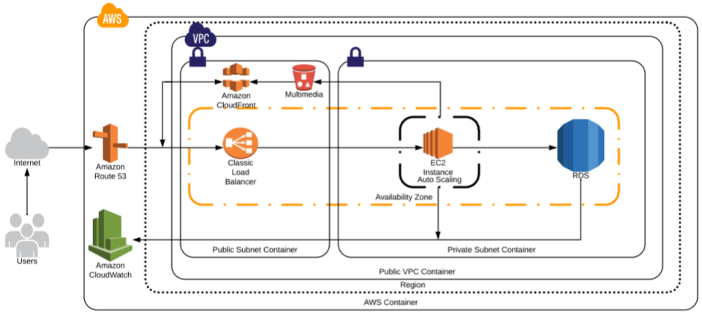
3-tiers architecture 2

source
https://medium.com/@codewithniraj/the-typical-3-tier-architecture-of-web-application-in-aws-f2f9d662fdfe



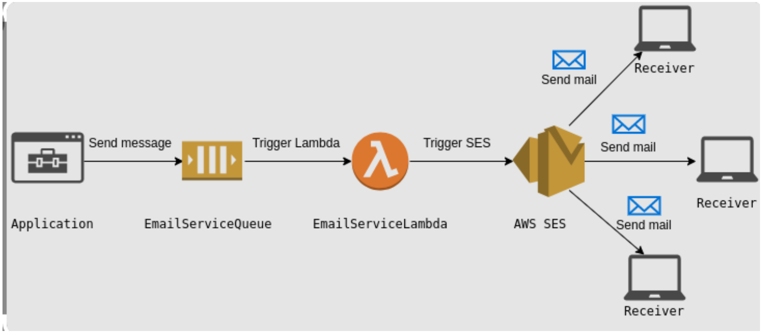
3-tiers architecture 3

source https://medium.com/@guillermo.velez/hosting-scalable-wordpres-aws-ff7ba6eec9ec



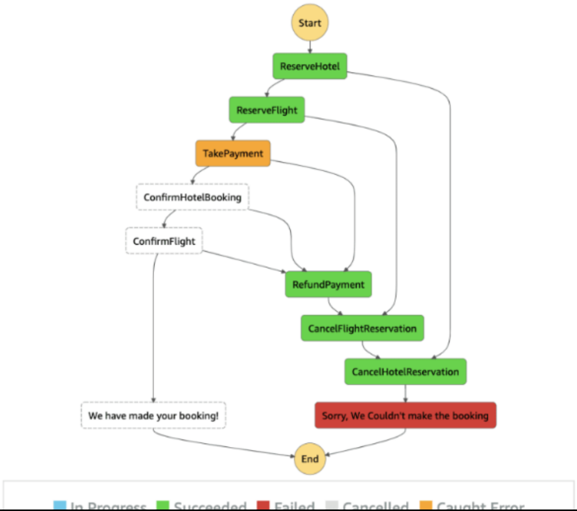
serverless 1

source https://levelup.gitconnected.com/implementing-an-e-mail-service-using-amazon-ses-7219440821de

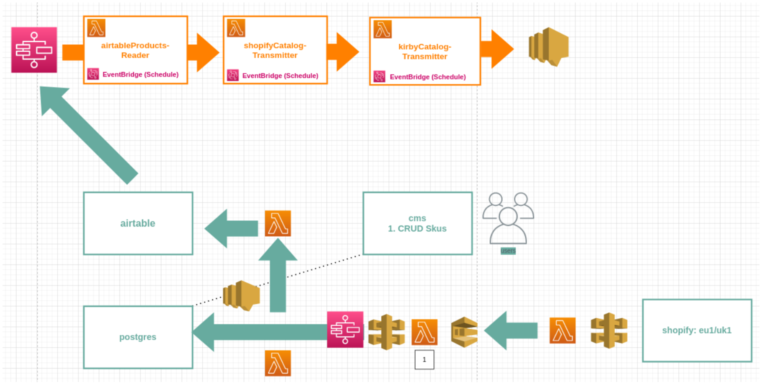


serverless 2: step functions

source `https://dev.to/cdkpatterns/learn-the-saga-stepfunction-pattern-today-single-table-dynamodb-lambdas-step-function-and-api-gateway-2o0a`



serverless 3: step functions



AWS Account/IAM/CLI

Best practices: Account and Groups

- Do not use admin account always
- Create separate account for AWS CLI
- Always create groups

Identity and Access Management

Identity and Access Management (IAM)

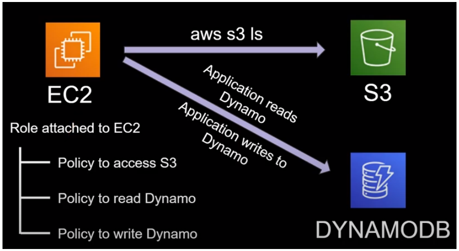
- IAM is a global service and is automatically available across ALL regions.
- after signign in, the email becomes your **root level account** and has full access.

Policies

- A policy is an object in AWS that defines permissions.
- It can be attached to groups/users/roles.

Roles

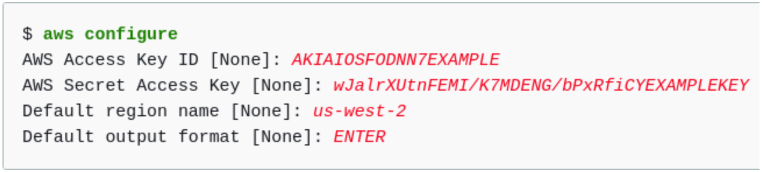
- An IAM role is an identity that can be attached to the **services**
- consists of policies
- important e.g. if you have same resources with different developer groups



Installation

AWS Command Line Interface (CLI)

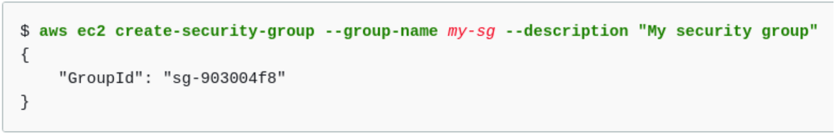
- Overview `https://docs.aws.amazon.com/cli/latest/userguide/install-cliv2-linux.html`
- the config data `/.aws/`



```
$ aws configure
AWS Access Key ID [None]: AKIAIOSFODNN7EXAMPLE
AWS Secret Access Key [None]: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
Default region name [None]: us-west-2
Default output format [None]: ENTER
```

Syntax

- aws [service] [command]
- my-sg is replaceable to your desired security group name
- The JSON document, including the curly braces, is output



```
$ aws ec2 create-security-group --group-name my-sg --description "My security group"
{
    "GroupId": "sg-903004f8"
}
```

Reference

- AWS CLI Command Reference `https://docs.aws.amazon.com/cli/latest/index.html`
- **create user** `https://docs.aws.amazon.com/cli/latest/reference/iam/create-user.html`
- **s3** `https://docs.aws.amazon.com/cli/latest/reference/s3/index.html`
- How to grant Access to S3 `https://aws.amazon.com/blogs/security/writing-iam-policies-how-to-grant-access-to-an-amazon`

Project 1/2

1. create group Content Manager and a user with only full access to S3
2. create another one with aws cli

Three-tiers Architecture

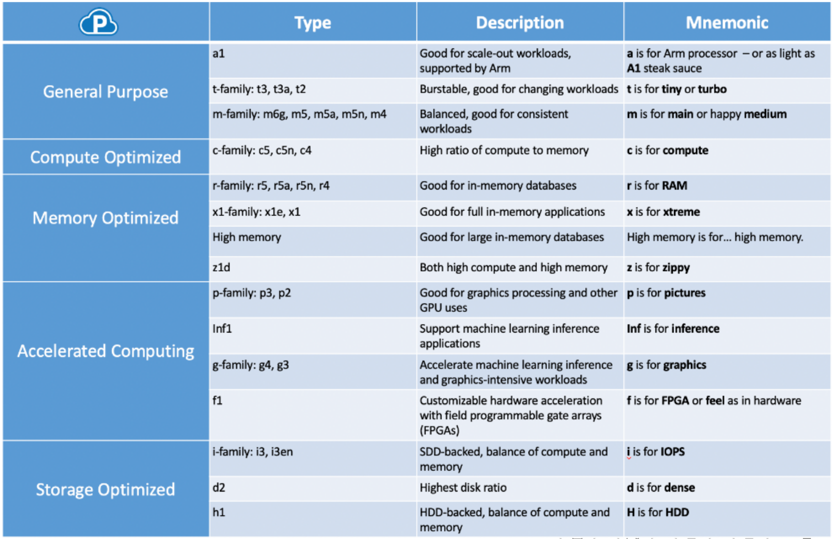
Types

An Overview `https://aws.amazon.com/ec2/instance-types/`

- around **400** instance types
- your account is limited to a maximum of 20 instances per EC2 region

EC2 Classification

source `https://www.parkmycloud.com/blog/ec2-instance-types/`



	Type	Description	Mnemonic
General Purpose	a1	Good for scale-out workloads, supported by Arm	**a** is for Arm processor – or as light as **A1** steak sauce
	t-family: t3, t3a, t2	Burstable, good for changing workloads	**t** is for **tiny** or **turbo**
	m-family: m6g, m5, m5a, m5n, m4	Balanced, good for consistent workloads	**m** is for **main** or happy **medium**
Compute Optimized	c-family: c5, c5n, c4	High ratio of compute to memory	**c** is for **compute**
Memory Optimized	r-family: r5, r5a, r5n, r4	Good for in-memory databases	**r** is for **RAM**
	x1-family: x1e, x1	Good for full in-memory applications	**x** is for **xtreme**
	High memory	Good for large in-memory databases	High memory is for… high memory.
	z1d	Both high compute and high memory	**z** is for **zippy**
Accelerated Computing	p-family: p3, p2	Good for graphics processing and other GPU uses	**p** is for **pictures**
	Inf1	Support machine learning inference applications	**Inf** is for **inference**
	g-family: g4, g3	Accelerate machine learning inference and graphics-intensive workloads	**g** is for **graphics**
	f1	Customizable hardware acceleration with field programmable gate arrays (FPGAs)	**f** is for **FPGA** or **feel** as in hardware
Storage Optimized	i-family: i3, i3en	SDD-backed, balance of compute and memory	**i** is for **IOPS**
	d2	Highest disk ratio	**d** is for **dense**
	h1	HDD-backed, balance of compute and memory	**H** is for **HDD**

AWS Compute Optimizer

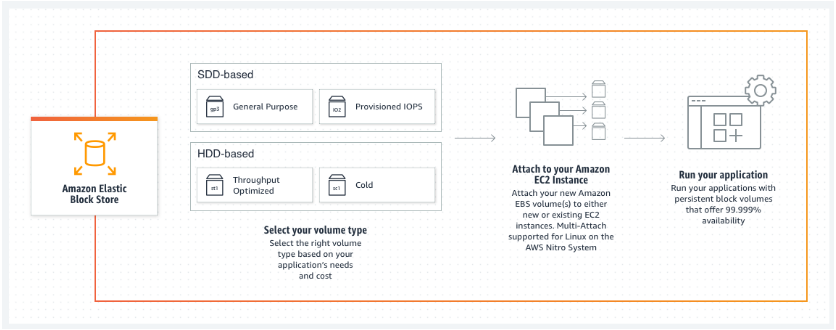
- `https://console.aws.amazon.com/compute-optimizer/`
- could be a life saver!
- identifies AWS **compute resources** as optimal/not optimal over a period of the last 14 days and offers reccomendations
- Pricing
  `https://aws.amazon.com/compute-optimizer/pricing/`
  (for 5 resources - USD 1.25 per month)

EBS Types

Amazon Elastic Block Store (EBS) is a scalable, high-performance block-storage service designed for Amazon Elastic Compute Cloud (Amazon EC2).
`https://aws.amazon.com/ebs/`

EBS Types

solid state drive (SSD) - fast and hard disk drive (HDD) - slow


Amazon Elastic Block Store
SDD-based
gp3
General Purpose
io2
Provisioned IOPS
HDD-based
st1
Throughput Optimized
sc1
Cold
Select your volume type
Select the right volume type based on your application's needs and cost
Attach to your Amazon EC2 Instance
Attach your new Amazon EBS volume(s) to either new or existing EC2 instances. Multi-Attach supported for Linux on the AWS Nitro System
Run your application
Run your applications with persistent block volumes that offer 99.999% availability

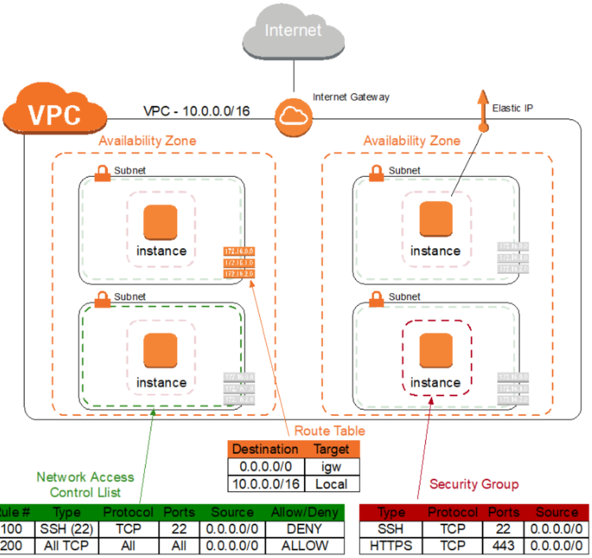
EBS Types

- more info https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-volume-types.html
- ESB optimization and compatibility https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-optimized.html
- best practice: calculate it before https://calculator.aws/

General info

- Virtual Private Cloud (VPC) is a logically isolated portion of the AWS cloud within a region
- Basic terminilogy: region, AZ, subnet, Route table, Igw, security group, NAL, Elastic IP
- Default: 5 VPs per region

VPC



Route Table

Destination	Target
0.0.0.0/0	igw
10.0.0.0/16	Local

Network Access Control Llist

Rule #	Type	Protocol	Ports	Source	Allow/Deny
100	SSH (22)	TCP	22	0.0.0.0/0	DENY
200	All TCP	All	All	0.0.0.0/0	ALLOW

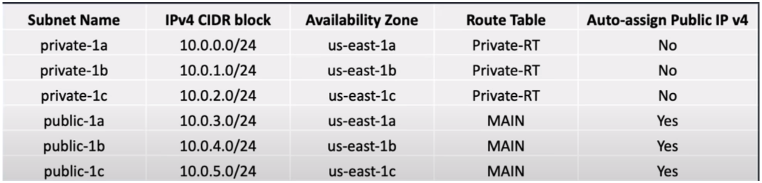
Security Group

Type	Protocol	Ports	Source
SSH	TCP	22	0.0.0.0/0
HTTPS	TCP	443	0.0.0.0/0

VPC Example

- explanation `https://docs.aws.amazon.com/vpc/latest/userguide/VPC_Subnets.html`
- When you create VPC, you have to specify Classless Inter-Domain Routing (CIDR) block, (e.g. 10.0.0.0/16)
- Best tool: `https://www.davidc.net/sites/default/subnets/subnets.html`
- tutorial `https://www.youtube.com/watch?v=z07HTSzzp3o`

VPC Examples



Subnet Name	IPv4 CIDR block	Availability Zone	Route Table	Auto-assign Public IP v4
private-1a	10.0.0.0/24	us-east-1a	Private-RT	No
private-1b	10.0.1.0/24	us-east-1b	Private-RT	No
private-1c	10.0.2.0/24	us-east-1c	Private-RT	No
public-1a	10.0.3.0/24	us-east-1a	MAIN	Yes
public-1b	10.0.4.0/24	us-east-1b	MAIN	Yes
public-1c	10.0.5.0/24	us-east-1c	MAIN	Yes

VPC Best practices

- Bigger CIDR blocks are typically better (more flexibility)
- Smaller subnets are OK for most use cases
- Split your resources in different AZs
- Application tiers per subnet (3 tiers - 3 subnets)

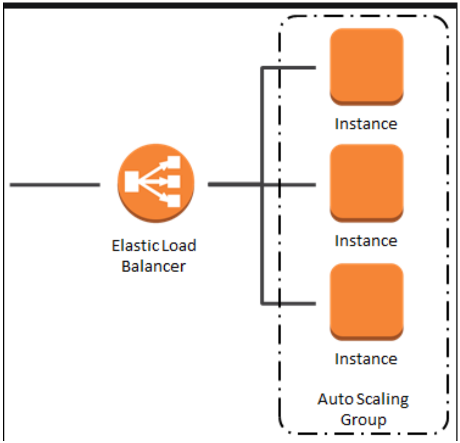
General info

- https://aws.amazon.com/rds/
- Pricing https://aws.amazon.com/rds/pricing/
- Free tier: 750 hours db.t2.micro (MySQL, MariaDB, PostgreSQL, Oracle BYOL or SQL Server) per month

Project 3/4

1. create and connect EC2 and RDS Mysql

General info

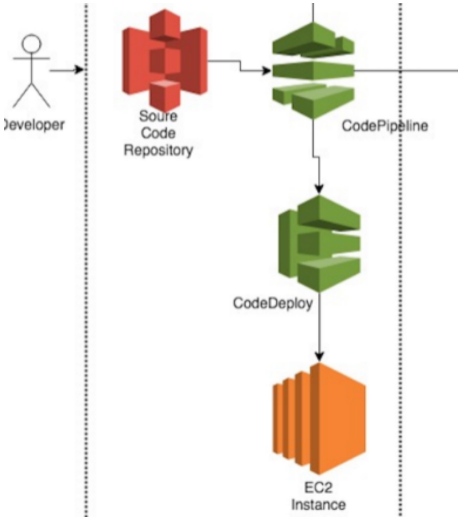


Deployment: Types

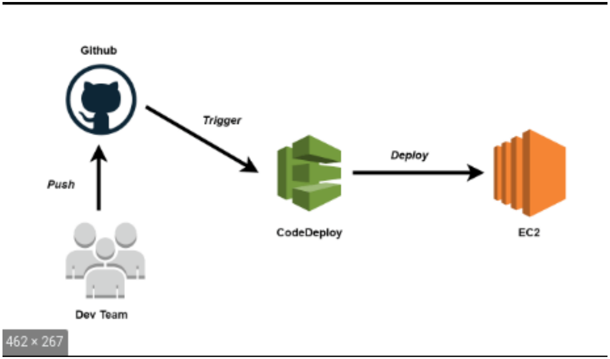
Deployment

- completely in AWS
- Bitbucket / GitLab / Github only
- hybrid version: e.g. Bitbucket and CodeDeploy

Deployment: AWS



Deployment: hybrid



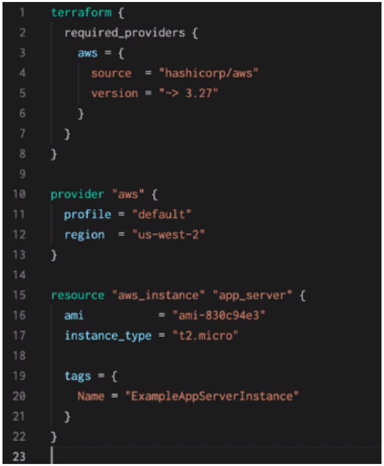
Provisioning

depends on the project and personal preferences

- aws-cli
- AWS Elastic Beanstalk `https://aws.amazon.com/elasticbeanstalk`: provis. and deploy.
- AWS CloudFormation `https://aws.amazon.com/cloudformation` provisioning and visualiz.
- AWS Cloud Development Kit https://aws.amazon.com/cdk IaC with familiar languages
- Terraform `https://www.terraform.io`

Terraform

HashiCorp Configuration Language (HCL)



```
terraform {
  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = "~> 3.27"
    }
  }
}

provider "aws" {
  profile = "default"
  region  = "us-west-2"
}

resource "aws_instance" "app_server" {
  ami           = "ami-830c94e3"
  instance_type = "t2.micro"

  tags = {
    Name = "ExampleAppServerInstance"
  }
}
```

My stack

- aws-cli
- Terraform
- Ansible `https://www.ansible.com/`
- Bitbucket
- CodeDeploy

Make your life easier...? -> Lightsail

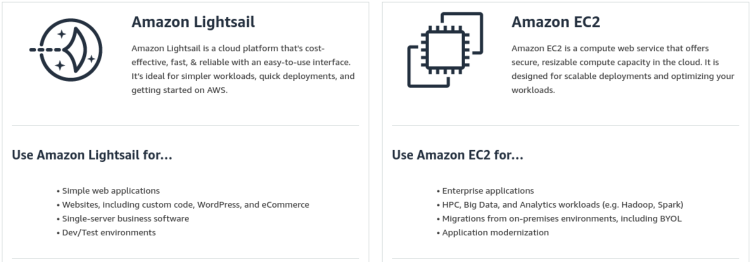
`https://aws.amazon.com/free/compute/lightsail/`

Basics

- It's less configurable than Amazon EC2, but it's easier to deploy and estimate costs.
- virtual machine, SSD-based storage, data transfer, DNS management, and a static IP are all offered as a package
- Bandwidth included in the price, no security groups to set up, no need to worry about EBS volumes sizing
- you can always migrate it to EC2 (take snapshot and export it)

When to use?

1. offical diff: `https://aws.amazon.com/free/compute/lightsail-vs-ec2/`



Amazon Lightsail

Amazon Lightsail is a cloud platform that's cost-effective, fast, & reliable with an easy-to-use interface. It's ideal for simpler workloads, quick deployments, and getting started on AWS.

Use Amazon Lightsail for...

- Simple web applications
- Websites, including custom code, WordPress, and eCommerce
- Single-server business software
- Dev/Test environments

Amazon EC2

Amazon EC2 is a compute web service that offers secure, resizable compute capacity in the cloud. It is designed for scalable deployments and optimizing your workloads.

Use Amazon EC2 for...

- Enterprise applications
- HPC, Big Data, and Analytics workloads (e.g. Hadoop, Spark)
- Migrations from on-premises environments, including BYOL
- Application modernization

Some Resources

- **blog** https://searchcloudcomputing.techtarget.com/tip/Compare-Amazon-Lightsail-vs-EC2-for-your-web-app-need
- **wordpress tutorial** https://aws.amazon.com/blogs/compute/deploying-a-highly-available-wordpress-site-on-amazon
- **run containers** https://aws.amazon.com/blogs/aws/lightsail-containers-an-easy-way-to-run-your-containers-in-the-cloud/

Lambda takes a step further

-> you do not care about low level machine

Basics

- `https://aws.amazon.com/lambda/`
- serveless microservices orchestration: **step functions** `https://aws.amazon.com/step-functions/`

Provisioning

- serverless `https://www.serverless.com/`
- terraform
- aws cdk
- terraform + serverless `https://www.serverless.com/blog/definitive-guide-terraform-serverless/`

Project 5

1. create a simple Lambda (trigger - API, Output - SNS E-Mail)

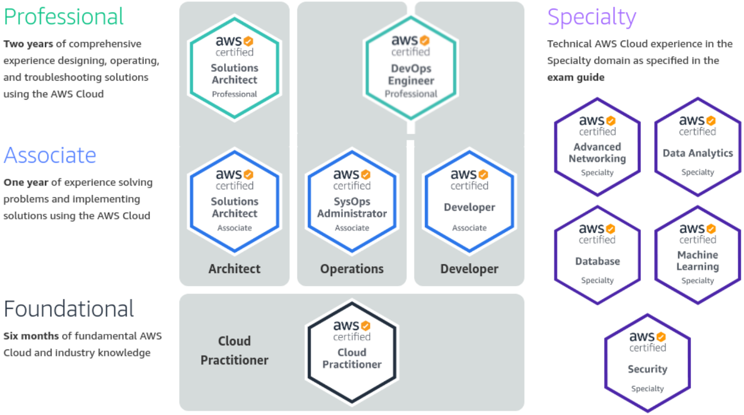
Outro

Further Steps

- Do not try to write a perfect architecture at first, try to **improve** it!
- Dive deeper in **one topic**, spend weeks till you have good understaning of it.
- AWS Certification

AWS Certification

Overview: `https://aws.amazon.com/certification/`



Professional

Two years of comprehensive experience designing, operating, and troubleshooting solutions using the AWS Cloud

Associate

One year of experience solving problems and implementing solutions using the AWS Cloud

Foundational

Six months of fundamental AWS Cloud and industry knowledge

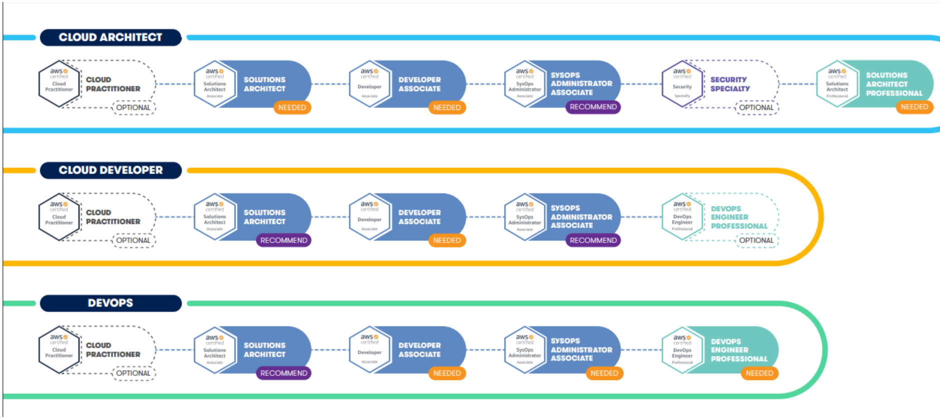
Specialty

Technical AWS Cloud experience in the Specialty domain as specified in the **exam guide**

aws certified Solutions Architect Professional
aws certified DevOps Engineer Professional
aws certified Solutions Architect Associate
aws certified SysOps Administrator Associate
aws certified Developer Associate
Architect
Operations
Developer
Cloud Practitioner
aws certified Cloud Practitioner
aws certified Advanced Networking Specialty
aws certified Data Analytics Specialty
aws certified Database Specialty
aws certified Machine Learning Specialty
aws certified Security Specialty

AWS Certification

Recommendations: https://go.acloudguru.com/rs/194-UHP-609/images/Cert-Guide-AWS-2020.pdf?ajs_aid=cbc821c3-421e-49dd-9ec0-381282abe674&_ga=2.110057989.1690562145.1639588668-1405476047.1637328720



AWS Certification

- valid for three years
- Costs
  - Practitioner = USD 100
  - Associate = USD 150
  - Professional = USD 300
  - Specialty = USD 300

AWS Learning

- Twitch https://aws.amazon.com/training/twitch/?sc_icampaign=aware_twitch_evergreen_free_midpage_cert_tnc_global_traincert&sc_ichannel=ha&sc_icontent=awssm-8747_tnc&sc_iplace=banner&trk=ha_awssm-8747_tnc&get-certified-vilt-courses-cards.sort-by=item.additionalFields.startDateSort&get-certified-vilt-courses-cards.sort-order=asc
- YoutTube
    - e.g. https://www.youtube.com/watch?v=ulprqHHWlng
    - Be a Better Dev https://www.youtube.com/channel/UCraiFqWi0qSIxXxXN4IHFBQ
- Cloud Guru
- Udemy
- Udacity

Thank you!

Questions? Suggestions?