— Use cases: Create Thumbnails for images uploaded onto S3. Run a serverless cron job

— API Gateway: Expose Lambda functions as HTTP API

→ Deploying and Managing Infrastructure at a Scale:

→ What is CloudFormation?

— Declarative way of outlining your AWS infrastructure for any resources.

— Eg: Within a CloudFormation template, you say:
- I want a security group
- I want two EC2 instances using this SG
- I want an S3 bucket
- I want a load balancer (ELB) in front of these machines.

— Then, CloudFormation creates those for you, in the right order, with the exact config that you specify.

→ Benefits of AWS CloudFormation.

1) Infrastructure as Code:
— No resources are manually created, which is excellent for control
— Changes to the infra are reviewed through code.

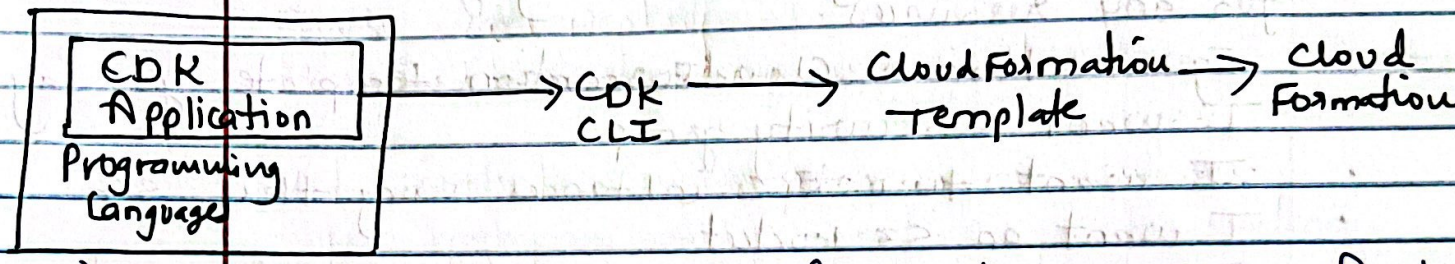
2) Cost:
— Each resources has cost tag attached.
— Can estimate the cost of your resources using CF template.
— Savings strategy: In Dev, you could automate deletion of templates at 5PM and recreate at 8pm safely.

3) Productivity:
— Able to destroy & re-create
— Automated generation of diagram for your templates
— Declarative programming.

4) Don't re-invent the wheel!
- Leverage existing templates on the web!
- Leverage the documentation.

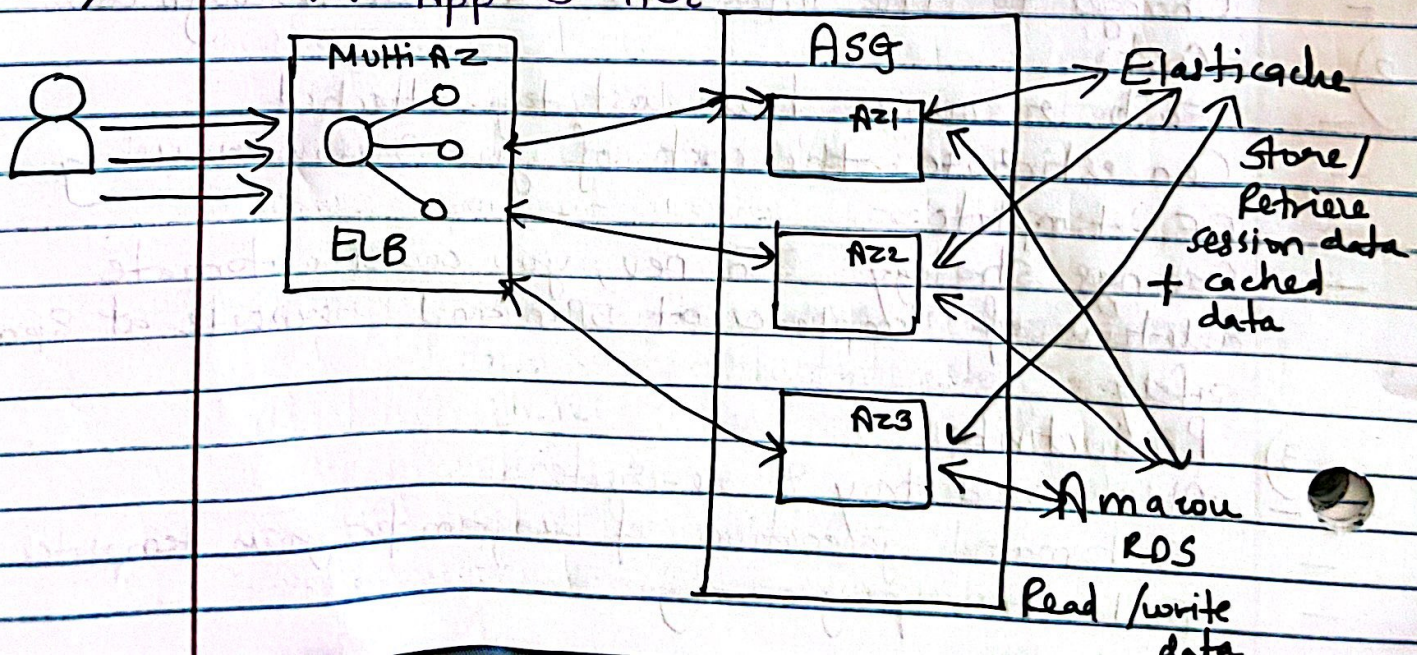
5) Supports (almost) all AWS resources, can "custom resources"!

→ AWS Cloud Development Kit (CDK)



1) Define your cloud infrastructure using a familiar language: Javascript/Typescript, Python, Java, .Net
2) Code is 'compiled' into a CloudFormation template (JSON / YAML).
3) Can therefore deploy infra and application runtime code together.
- Great for lambda functions
- Great for Docker containers in ECS / EKS

→ Web App 3-tier:



→ Developer problems on AWS:

1) Managing infrastructure
2) Deploying code
3) Configurable all the databases, load balances, etc
4) Scaling concerns
5) Most web apps have the same architecture (ALB + ASG)

→ AWS Elastic Beanstalk (ELB)

- Developer centric view of deploying an application on AWS.
- Uses all components.
- All in one view.
- Full control over the config.
- Beanstalk = Platform as a Service (PaaS).
- Free but pay for the underlying instances.
- Managed service:
  1) Instance config / os handled by Beanstalk.
  2) Deployment strategy is configurable but performed by EB.
  3) Capacity provisioning
  4) LB & ASG
  5) Application health monitoring & responsiveness.

Just the application code is the responsibility of the developer.

- Three architecture models:
  1) Single Instance Deployment: Good for dev.
  2) LB + ASG: Great for production or pre-production web applications
  3) ASG only: Great for non-web apps in production
- Support for many platforms:
  - Go
  - Java SE
  - Java with Tomcat
  - .NET on Windows with IIS
  - Node.js
  - PHP
  - Python
  - Ruby
  - Packer Builder

- Singled Container Docker
- Multi-Container Docker
- Preconfigured Docker

— If not supported, you can write your custom platform (advanced)

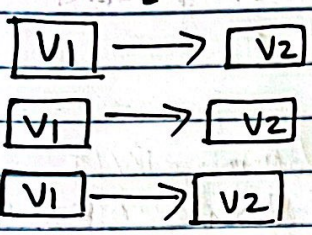
→ Elastic Beanstalk : Health Monitoring.

— Health agent pushes metrics to cloudWatch

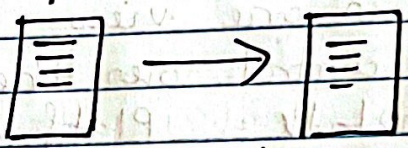
— Checks for app health, publishes health events

→ AWS CodeDeploy

EC2 Instances being upgraded.



On premises servers being upgraded



— Deploy your application automatically

— Works on EC2 instances

— Works with on-Premise servers

— Hybrid service.

— Servers / Instances must be provisioned and configured ahead of time with the CodeDeploy agent

→ AWS CodeCommit

— Code needs to be stored before pushing to the server.

— Codecommit (same like Github)

— Source-control service that hosts Git based repos.

— Easy to collaborate with others on code

— Code changes are automatically versioned.

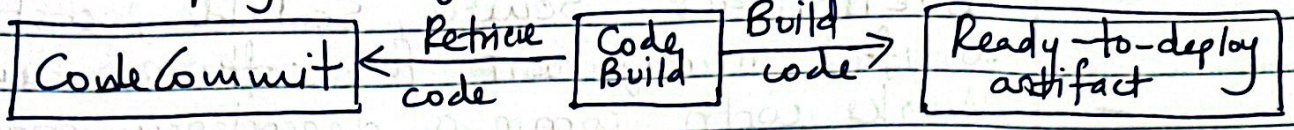
— Benefits:

1) Fully managed
2) Scalable & highly available
3) Private, secured, Integrated with AWS.

AWS CodeBuild

Code building service in the cloud

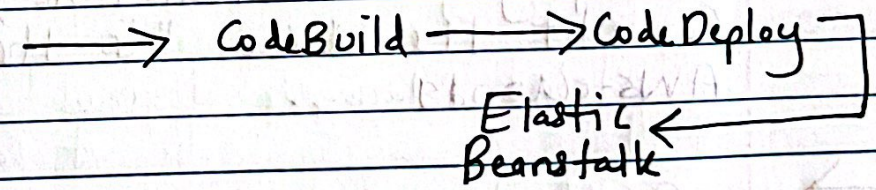
- Compiles source code, run tests, and produces packages that are ready to be deployed (by CodeDeploy for eg.).



- Benefits:
  1) Fully managed, serverless
  2) Continuously scalable & highly available
  3) Secure
  4) Pay-as-you-go pricing — only pay for the build time.

AWS CodePipeline

3. CodePipeline: orchestration layer



- Orchestrate the different steps to have the code automatically pushed to production.
- Code ⇒ Build ⇒ Test ⇒ Provision ⇒ Deploy
- Basis for CICD
- Benefits:
- Fully managed, compatible with CodeCommit, CodeBuild, CodeDeploy, Elastic Beanstalk, CloudFormation, Github, ...
- Fast delivery & rapid updates.

AWS CodeArtifact

- Software packages depend on each other to be built (also called code dependencies).
- Storing & retrieving these dependencies is called 'artifact management'.
- CodeArtifact: Secure, Scalable, cost-effective artifact management for software development.
- Works with common dependency management tools such as Maven, Gradle, npm, yarn, twine, pip.
- Developers and CodeBuild can then retrieve dependencies straight from CodeArtifact.

AWS Codestar

- Unified UI to easily manage software development activities in one place.
- "Quick way" to get started to correctly set-up CodeCommit, CodePipeline, CodeBuild, CodeDeploy, Elastic Beanstalk, EC2 etc.
- Can edit the code "in-the-cloud" using AWS Cloud9.

AWS Cloud9

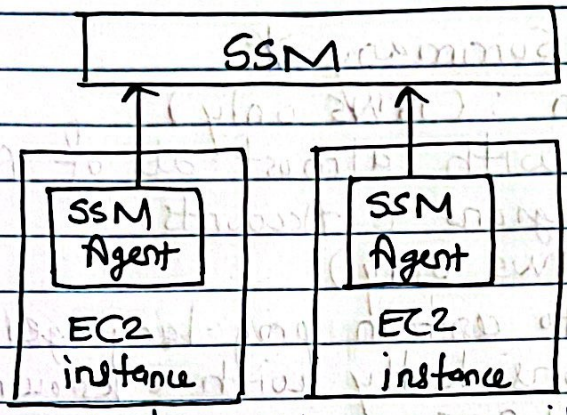
- Cloud IDE for writing, running & debugging code.
- "Classic" IDE (IntelliJ, Vs Code) are downloaded on a computer before being used.
- Cloud IDE can be used within a web browser.
- Also allows for code collaboration in real-time (pair programming).

AWS Systems Manager (SSM)

- Helps you manage your EC2 and on-premises systems at scale.
- Hybrid AWS Service.
- Get operational insights about the state of your

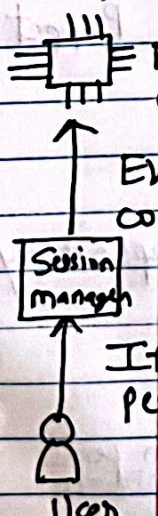
infrastructure
- Suite of 10+ products
- Most important features:
  1) Patching automation for enhanced compliance.
  2) Run commands across an entire fleet of servers.
  3) Store parameter config with the SSM Parameter Store.
- Both for Window, Linux OS.

→ How Systems Manager works?



- Need to install the SSM agent onto the systems we control.
- Installed by default on Amazon Linux AMI & some Ubuntu AMI.
- If instance can't be controlled with SSM, it's probably an issue with the SSM agent.
- Run commands, patch & configure our servers.

→ Systems Manager - SSM Session Manager



- Start a secure shell on your EC2 and on-premise servers.
- No SSH access
- No Bastion hosts
- No SSH keys needed
- Supports Linux, MacOS, Windows
- Sends session log data to S3 or CloudWatch logs.

→ AWS OpsWork

- Chef & Puppet - Help perform server config automatically or repetitive actions.
- Work great with EC2 & on-premise VM
- OpsWork = Managed Chef & Puppet
- Alternative to SSM.
- Only provision standard AWS resources: EC2 instances, Databases, LBs, EBS volumes
- Chef & Puppet needed ⇒ AWS OpsWorks

* Deployment - Summary *

- CloudFormation : (AWS only)
  1) IaaC, works with almost all of AWS resources
  2) Repeat across Regions & Accounts.
- Beanstalk: (AWS only)
  1) PaaS, limited to certain prog. languages or Docker
  2) Deploy code consistently with a known architecture ex. ALB + EC2 + RDS
  3) - CodeDeploy : Hybrid; deploy & upgrade any application onto servers.
- Systems Manager: Hybrid: Patch, configure and run commands at scale
- OpsWorks: Hybrid : Managed Chef and Puppet in AWS.

* Developer Services - Summary *

- CodeCommit: Store code in private git repository (version controlled).
- CodeBuild: Build & test code in AWS
- CodeDeploy: Deploy code onto servers
- CodePipeline: Orchestration of pipeline (from code to build to deploy)
- CodeArtifact: Store software packages/ dependencies on AWS.
- CodeStar : Unified view for allowing developers to do CICD and code

~~Coding~~ Cloudg : Cloud IDE

AWS CDK : Define your cloud infrastructure using a programming language.