master | 5 Branches | 0 Tags | Go to file | Go to file | Code

0xflotus fix: small error (#770) cde0294 · 4 years ago

| | | |
|---|---|---|
| .github/workflows | Fix broken links (#762) | 6 years ago |
| admin | Update authors (and separate roles into groups fo... | 9 years ago |
| figures | Update diagram for data transfer (#485) | 9 years ago |
| translations | Russian Translation (#1) (#744) | 6 years ago |
| .travis.yml | Whitelist swagger.io due to bug | 8 years ago |
| AUTHORS.md | Update credits. | 8 years ago |
| CODE_OF_CONDUCT.md | Create CODE_OF_CONDUCT.md | 9 years ago |
| CONTRIBUTING.md | Fix signup link. | 10 years ago |
| LICENSE.txt | Added cc-by-4.0 license, issue #535 | 9 years ago |
| README.md | fix: small error (#770) | 4 years ago |


# The Open Guide to Amazon Web Services

Chat Slack ⇦ Join us!

Credits · Contributing guidelines

## Table of Contents

- Managing Servers and Applications

| Specific AWS Services | Basics | Tips | Gotchas |
|---|---|---|---|
| ALB | 📗 | 📘 | 📕 |
| AMIs | 📗 | 📘 | 📕 |
| API Gateway | 📗 | 📘 | 📕 |
| Auto Scaling | 📗 | 📘 | 📕 |
| Batch | 📗 | 📘 | |
| Certificate Manager | 📗 | 📘 | 📕 |
| CLB (ELB) | 📗 | 📘 | 📕 |
| CloudFront | 📗 | 📘 | 📕 |
| CloudFormation | 📗 | 📘 | 📕 |
| CloudWatch | 📗 | 📘 | 📕 |
| Device Farm | 📗 | 📘 | 📕 |
| DirectConnect | 📗 | 📘 | |
| DynamoDB | 📗 | 📘 | 📕 |
| EBS | 📗 | 📘 | 📕 |
| EC2 | 📗 | 📘 | 📕 |
| ECS | 📗 | 📘 | |
| EKS | 📗 | 📘 | 📕 |
| EFS | 📗 | 📘 | 📕 |

| Specific AWS Services | Basics | Tips | Gotchas |
|---|---|---|---|
| Elastic Beanstalk | 📗 | 📘 | 📕 |
| Elastic IPs | 📗 | 📘 | 📕 |
| ElastiCache | 📗 | 📘 | 📕 |
| EMR | 📗 | 📘 | 📕 |
| Fargate | 📗 | 📘 | 📕 |
| Glacier | 📗 | 📘 | 📕 |
| IoT | 📗 | 📘 | 📕 |
| Kinesis Firehose | | | 📕 |
| Kinesis Streams | 📗 | 📘 | 📕 |
| KMS | 📗 | 📘 | 📕 |
| Lambda | 📗 | 📘 | 📕 |
| Load Balancers | 📗 | 📘 | 📕 |
| Mobile Hub | 📗 | 📘 | 📕 |
| OpsWorks | 📗 | 📘 | 📕 |
| Quicksight | 📗 | | 📕 |
| RDS | 📗 | 📘 | 📕 |
| RDS Aurora | 📗 | 📘 | 📕 |
| RDS Aurora MySQL | 📗 | 📘 | 📕 |
| RDS Aurora PostgreSQL | 📗 | 📘 | 📕 |

| Specific AWS Services | Basics | Tips | Gotchas |
| --- | --- | --- | --- |
| RDS MySQL and MariaDB | 📗 | 📘 | 📕 |
| RDS PostgreSQL | 📗 | 📘 | 📕 |
| RDS SQL Server | 📗 | 📘 | 📕 |
| Redshift | 📗 | 📘 | 📕 |
| Route 53 | 📗 | 📘 | 📕 |
| S3 | 📗 | 📘 | 📕 |
| Security and IAM | 📗 | 📘 | 📕 |
| SES | 📗 | 📘 | 📕 |
| SNS | 📗 | 📘 | 📕 |
| SQS | 📗 | 📘 | 📕 |
| Step Functions | 📗 | 📘 | 📕 |
| WAF | 📗 | 📘 | 📕 |
| VPCs, Network Security, and Security Groups | 📗 | 📘 | 📕 |

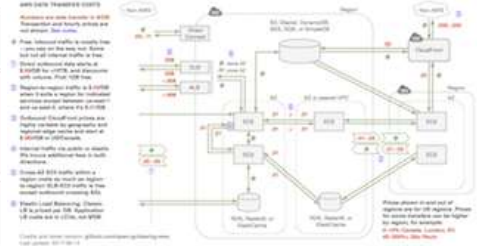
Special Topics

- High Availability
- Billing and Cost Management
- Further Reading

Legal

- Disclaimer
- License

Figures and Tables



- Figure: Tools and Services Market Landscape: A selection of third-party companies/products
- Figure: AWS Data Transfer Costs: Visual overview of data transfer costs
- Table: Service Matrix: How AWS services compare to alternatives
- Table: AWS Product Maturity and Releases: AWS product releases
- Table: Storage Durability, Availability, and Price: A quantitative comparison

## Why an Open Guide?

A lot of information on AWS is already written. Most people learn AWS by reading a blog or a "getting started guide" standard AWS references. Nonetheless, trustworthy and practical information and recommendations aren't easy to co documentation is a great but sprawling resource few have time to read fully, and it doesn't include anything but offic experiences of engineers. The information in blogs or Stack Overflow is also not consistently up to date.

This guide is by and for engineers who use AWS. It aims to be a useful, living reference that consolidates links, tips, g It arose from discussion and editing over beers by several engineers who have used AWS extensively.

Before using the guide, please read the **license** and **disclaimer**.

Back to top ⬆️

### Please help!

omissions or errors.


Please help by joining the Slack channel (we like to talk about AWS in general, even if you only have questions — di community and guides improvements) and contributing to the guide. This guide is *open to contributions*, so unlike a improving. Like any open source effort, we combine efforts but also review to ensure high quality.

## Scope

- Currently, this guide covers selected "core" services, such as EC2, S3, Load Balancers, EBS, and IAM, and partial d other services. We expect it to expand.
- It is not a tutorial, but rather a collection of information you can read and return to. It is for both beginners and
- The goal of this guide is to be:
  - **Brief:** Keep it dense and use links
  - **Practical:** Basic facts, concrete details, advice, gotchas, and other "folk knowledge"
  - **Current:** We can keep updating it, and anyone can contribute improvements
  - **Thoughtful:** The goal is to be helpful rather than present dry facts. Thoughtful opinion with rationale is welc and opinions based on real experience can be extremely valuable. (We believe this is both possible with a g in some other venues.)
- This guide is not sponsored by AWS or AWS-affiliated vendors. It is written by and for engineers who use AWS.

## Legend

- 📒 Marks standard/official AWS pages and docs
- 🔹 Important or often overlooked tip
- ❗ "Serious" gotcha (used where risks or time or resource costs are significant: critical security risks, mistakes wi or poor architectural choices that are fundamentally difficult to correct)
- 🔸 "Regular" gotcha, limitation, or quirk (used where consequences are things not working, breaking, or not sca

- 📜 Undocumented feature (folklore)
- 🐥 Relatively new (and perhaps immature) services or features
- ⏱️ Performance discussions
- ⛓️ Lock-in: Products or decisions that are likely to tie you to AWS in a new or significant way — that is, later mo alternative would be costly in terms of engineering effort
- 🚪 Alternative non-AWS options
- 💸 Cost issues, discussion, and gotchas
- 🕰️ A mild warning attached to "full solution" or opinionated frameworks that may take significant time to unde your needs exactly; the opposite of a point solution (the cathedral is a nod to Raymond's metaphor)
- 🟩 🟦 🟥 Colors indicate basics, tips, and gotchas, respectively.
- 🚧 Areas where correction or improvement are needed (possibly with link to an issue — do help!)

## General Information

### When to Use AWS

- AWS is the dominant public cloud computing provider.
  - In general, "cloud computing" can refer to one of three types of cloud: "public," "private," and "hybrid." AWS since anyone can use it. Private clouds are within a single (usually large) organization. Many companies use public clouds.
  - The core features of AWS are infrastructure-as-a-service (IaaS) — that is, virtual machines and supporting in service models include platform-as-a-service (PaaS), which typically are more fully managed services that de applications, or software-as-a-service (SaaS), which are cloud-based applications. AWS does offer a few pro other models, too.
  - In business terms, with infrastructure-as-a-service you have a variable cost model — it is OpEx, not CapEx (t purchased contracts are still CapEx).
- AWS's TTM revenue was **$37.549 billion** as of Q1 2020 according to their earnings results (slide 14 in the linked Amazon.com's total revenue (slide 11 in the same deck) for the same TTM period.
- **Main reasons to use AWS:**

    - If your company is building systems or products that may need to scale
    - and you have technical know-how
    - and you want the most flexible tools
    - and you're not significantly tied into different infrastructure already
    - and you don't have internal, regulatory, or compliance reasons you can't use a public cloud-based solution
    - and you're not on a Microsoft-first tech stack
    - and you don't have a specific reason to use Google Cloud
    - and you can afford, manage, or negotiate its somewhat higher costs
    - ... then AWS is likely a good option for your company.
- Each of those reasons above might point to situations where other services are preferable. In practice, many, if n well as a number of modern large companies can or already do benefit from using AWS. Many large enterprises internal infrastructure to Azure, Google Cloud, and AWS.
- **Costs:** Billing and cost management are such big topics that we have an entire section on this.
- 🔹 **EC2 vs. other services:** Most users of AWS are most familiar with EC2, AWS' flagship virtual server product, ar like S3 and CLBs. But AWS products now extend far beyond basic IaaS, and often companies do not properly un the many AWS services and how they can be applied, due to the sharply growing number of services, their nove branding confusion, and fear of ⛓️ lock-in to proprietary AWS technology. Although a bit daunting, it's importar makers in companies to understand the breadth of the AWS services and make informed decisions. (We hope th
- 🟧 **AWS vs. other cloud providers:** While AWS is the dominant IaaS provider (31% market share in this 2016 esti competition and alternatives that are better suited to some companies. This Gartner report has a good overview :
    - Google Cloud Platform. GCP arrived later to market than AWS, but has vast resources and is now used wide including a few large ones. It is gaining market share. Not all AWS services have similar or analogous service particular, GCP offers some more advanced machine learning-based services like the Vision, Speech, and Na not common to switch once you're up and running, but it does happen: Spotify migrated from AWS to Goo discussion on Quora about relative benefits. Of particular note is that VPCs in GCP are global by default wit while AWS' VPCs have to live within a particular region. This gives GCP an edge if you're designing applicati from the beginning. It's also possible to share one GCP VPC between multiple projects (roughly analogous t AWS you'd have to peer them. It's also possible to peer GCP VPCs in a similar manner to how it's done in AW

    - Microsoft Azure is the de facto choice for companies and teams that are focused on a Microsoft stack, and significant emphasis on Linux as well
    - In **China**, AWS' footprint is relatively small. The market is dominated by Alibaba's Alibaba Cloud, formerly ca
    - Companies at (very) large scale may want to reduce costs by managing their own infrastructure. For exampl their own infrastructure.
    - Other cloud providers such as Digital Ocean offer similar services, sometimes with greater ease of use, more lower cost. However, none of these match the breadth of products, mind-share, and market domination AW
    - Traditional managed hosting providers such as Rackspace offer cloud solutions as well.
- 🚪 **AWS vs. PaaS:** If your goal is just to put up a single service that does something relatively simple, and you're managing operations engineering, consider a platform-as-a-service such as Heroku. The AWS approach to PaaS arguably more complex, especially for simple use cases.
- 🚪 **AWS vs. web hosting:** If your main goal is to host a website or blog, and you don't expect to be building an a service, you may wish consider one of the myriad web hosting services.
- 🚪 **AWS vs. managed hosting:** Traditionally, many companies pay managed hosting providers to maintain physi build and deploy their software on top of the rented hardware. This makes sense for businesses who want direct due to legacy, performance, or special compliance constraints, but is usually considered old fashioned or unnece centric startups and younger tech companies.
- **Complexity:** AWS will let you build and scale systems to the size of the largest companies, but the complexity of scale requires significant depth of knowledge and experience. Even very simple use cases often require more kn AWS than in a simpler environment like Heroku or Digital Ocean. (This guide may help!)
- **Geographic locations:** AWS has data centers in over a dozen geographic locations, known as **regions**, in Europe South America, and now Australia and India. It also has many more **edge locations** globally for reduced latency
    - See the current list of regions and edge locations, including upcoming ones.
    - If your infrastructure needs to be in close physical proximity to another service for latency or throughput re to an ad exchange), viability of AWS may depend on the location.
- ⛓️ **Lock-in:** As you use AWS, it's important to be aware when you are depending on AWS services that do not ha
    - Lock-in may be completely fine for your company, or a significant risk. It's important from a business perspe explicitly, and consider the cost, operational, business continuity, and competitive risks of being tied to AWS

and reliable vendor, many companies are comfortable with using AWS to its full extent. Others can tell stori
"cloud jail" when costs spiral.
    - Generally, the more AWS services you use, the more lock-in you have to AWS — that is, the more engineeri
money) it will take to change to other providers in the future.
    - Basic services like virtual servers and standard databases are usually easy to migrate to other providers or o
load balancers and IAM are specific to AWS but have close equivalents from other providers. The key thing
engineers are architecting systems around specific AWS services that are not open source or relatively inter
Lambda, API Gateway, Kinesis, Redshift, and DynamoDB do not have substantially equivalent open source o
equivalents, while EC2, RDS (MySQL or Postgres), EMR, and ElastiCache more or less do. (See more below, w
⛓.)
- **Combining AWS and other cloud providers:** Many customers combine AWS with other non-AWS services. For e
secure data might be in a managed hosting provider, while other systems are AWS. Or a company might only us
doing everything else. However small startups or projects starting fresh will typically stick to AWS or Google Clo
- **Hybrid cloud:** In larger enterprises, it is common to have hybrid deployments encompassing private cloud or on
— or other enterprise cloud providers like IBM/Bluemix, Microsoft/Azure, NetApp, or EMC.
- **Major customers:** Who uses AWS and Google Cloud?
    - AWS's list of customers includes large numbers of mainstream online properties and major brands, such as
(moving to Google Cloud), Airbnb, Expedia, Yelp, Zynga, Comcast, Nokia, and Bristol-Myers Squibb.
    - Azure's list of customers includes companies such as NBC Universal, 3M and Honeywell Inc.
    - Google Cloud's list of customers is large as well, and includes a few mainstream sites, such as Snapchat, Bes
Music.

Back to top ⬆️

## Which Services to Use

- AWS offers a *lot* of different services — about a hundred at last count.
- Most customers use a few services heavily, a few services lightly, and the rest not at all. What services you'll use
Choices differ substantially from company to company.

- **Immature and unpopular services:** Just because AWS has a service that sounds promising, it doesn't mean you services are very narrow in use case, not mature, are overly opinionated, or have limitations, so building your ow We try to give a sense for this by breaking products into categories.
- **Must-know infrastructure:** Most typical small to medium-size users will focus on the following services first. If y systems, you likely need to know at least a little about all of these. (Even if you don't use them, you should learn choice intelligently.)
  - IAM: User accounts and identities (you need to think about accounts early on!)
  - EC2: Virtual servers and associated components, including:
    - AMIs: Machine Images
    - Load Balancers: CLBs and ALBs
    - Autoscaling: Capacity scaling (adding and removing servers based on load)
    - EBS: Network-attached disks
    - Elastic IPs: Assigned IP addresses
  - S3: Storage of files
  - Route 53: DNS and domain registration
  - VPC: Virtual networking, network security, and co-location; you automatically use
  - CloudFront: CDN for hosting content
  - CloudWatch: Alerts, paging, monitoring
- **Managed services:** Existing software solutions you could run on your own, but with managed deployment:
  - RDS: Managed relational databases (managed MySQL, Postgres, and Amazon's own Aurora database)
  - EMR: Managed Hadoop
  - Elasticsearch: Managed Elasticsearch
  - ElastiCache: Managed Redis and Memcached
- **Optional but important infrastructure:** These are key and useful infrastructure components that are less widely have legitimate reasons to prefer alternatives, so evaluate with care to be sure they fit your needs:
  - ⛓ Lambda: Running small, fully managed tasks "serverless"
  - CloudTrail: AWS API logging and audit (often neglected but important)
  - ⛓ 🏗 CloudFormation: Templatized configuration of collections of AWS resources

    - 👩‍🚒 Elastic Beanstalk: Fully managed (PaaS) deployment of packaged Java, .NET, PHP, Node.js, Python, Ruby, applications
    - 🐥 EFS: Network filesystem compatible with NFSv4.1
    - ⛓️ 👩‍🚒 ECS: Docker container/cluster management (note Docker can also be used directly, without ECS)
    - 👩‍🚒 EKS: Kubernetes (K8) Docker Container/Cluster management
    - ⛓️ ECR: Hosted private Docker registry
    - 🐥 Config: AWS configuration inventory, history, change notifications
    - 🐥 X-Ray: Trace analysis and debugging for distributed applications such as microservices.
- **Special-purpose infrastructure:** These services are focused on specific use cases and should be evaluated if they Many also are proprietary architectures, so tend to tie you to AWS.
    - ⛓️ DynamoDB: Low-latency NoSQL key-value store
    - ⛓️ Glacier: Slow and cheap alternative to S3
    - ⛓️ Kinesis: Streaming (distributed log) service
    - ⛓️ SQS: Message queueing service
    - ⛓️ Redshift: Data warehouse
    - 🐥 QuickSight: Business intelligence service
    - SES: Send and receive e-mail for marketing or transactions
    - ⛓️ API Gateway: Proxy, manage, and secure API calls
    - ⛓️ IoT: Manage bidirectional communication over HTTP, WebSockets, and MQTT between AWS and clients "things" like appliances or sensors)
    - ⛓️ WAF: Web firewall for CloudFront to deflect attacks
    - ⛓️ KMS: Store and manage encryption keys securely
    - Inspector: Security audit
    - Trusted Advisor: Automated tips on reducing cost or making improvements
    - 🐥 Certificate Manager: Manage SSL/TLS certificates for AWS services
    - 🐥 ⛓️ Fargate: Docker containers management, backend for ECS and EKS
- **Compound services:** These are similarly specific, but are full-blown services that tackle complex problems and m depends on your requirements. If you have large or significant need, you may have these already managed by in

engineering teams.
    - Machine Learning: Machine learning model training and classification
    - Lex: Automatic speech recognition (ASR) and natural language understanding (NLU)
    - Polly: Text-to-speech engine in the cloud
    - Rekognition: Service for image recognition
    - ⛓️🕵️ Data Pipeline: Managed ETL service
    - ⛓️🕵️ SWF: Managed state tracker for distributed polyglot job workflow
    - ⛓️🕵️ Lumberyard: 3D game engine
- **Mobile/app development:**
    - SNS: Manage app push notifications and other end-user notifications
    - ⛓️🕵️ Cognito: User authentication via Facebook, Twitter, etc.
    - Device Farm: Cloud-based device testing
    - Mobile Analytics: Analytics solution for app usage
    - 🕵️ Mobile Hub: Comprehensive, managed mobile app framework
- **Enterprise services:** These are relevant if you have significant corporate cloud-based or hybrid needs. Many sma
startups use other solutions, like Google Apps or Box. Larger companies may also have their own non-AWS IT sc
    - AppStream: Windows apps in the cloud, with access from many devices
    - Workspaces: Windows desktop in the cloud, with access from many devices
    - WorkDocs (formerly Zocalo): Enterprise document sharing
    - WorkMail: Enterprise managed e-mail and calendaring service
    - Directory Service: Microsoft Active Directory in the cloud
    - Direct Connect: Dedicated network connection between office or data center and AWS
    - Storage Gateway: Bridge between on-premises IT and cloud storage
    - Service Catalog: IT service approval and compliance
- **Probably-don't-need-to-know services:** Bottom line, our informal polling indicates these services are just not br
for good reasons:
    - Snowball: If you want to ship petabytes of data into or out of Amazon using a physical appliance, read on.
    - Snowmobile: Appliances are great, but if you've got exabyte scale data to get into Amazon, nothing beats a

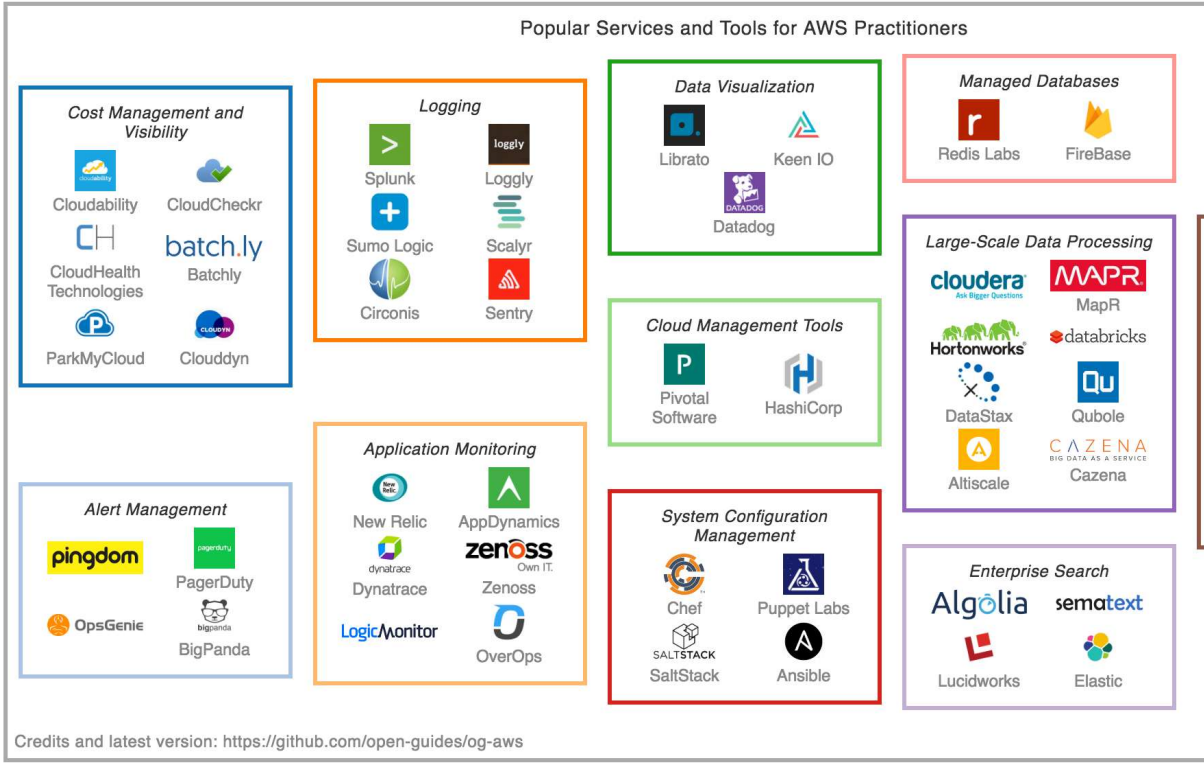
    - CodeCommit: Git service. You're probably already using GitHub or your own solution (Stackshare has inform
    - 👩‍🚀 CodePipeline: Continuous integration. You likely have another solution already.
    - 👩‍🚀 CodeDeploy: Deployment of code to EC2 servers. Again, you likely have another solution.
    - 👩‍🚀 OpsWorks: Management of your deployments using Chef or (as of November 2017) Puppet Enterprise.
- AWS in Plain English offers more friendly explanation of what all the other different services are.

Back to top ⬆️

## Tools and Services Market Landscape

There are now enough cloud and "big data" enterprise companies and products that few can keep up with the marke

We've assembled a landscape of a few of the services. This is far from complete, but tries to emphasize services that
practitioners — services that specifically help with AWS, or a complementary, or tools almost anyone using AWS mus




🚧 *Suggestions to improve this figure? Please file an issue.*

Back to top ⬆️

## Common Concepts

- 📒 The AWS General Reference covers a bunch of common concepts that are relevant for multiple services.
- AWS allows deployments in regions, which are isolated geographic locations that help you reduce latency or off
  Regions contain availability zones(AZs), which are typically the first tool of choice for high availability). AZs are p

another even within the same region, and may span multiple physical data centers. While they are connected via disasters afflicting one should not affect others.

- Each service has API **endpoints** for each region. Endpoints differ from service to service and not all services are a listed in these tables.
- **Amazon Resource Names (ARNs)** are specially formatted identifiers for identifying resources. They start with 'ar services, and in particular for IAM policies.

Back to top ⬆️

## Service Matrix

Many services within AWS can at least be compared with Google Cloud offerings or with internal Google services. An assemble the same thing yourself with open source software. This table is an effort at listing these rough correspond table is imperfect as in almost every case there are subtle differences of features!)

| Service | AWS | Google Cloud | Google Internal | Microsoft Azure | Other providers | Ope "bu |
|---|---|---|---|---|---|---|
| Virtual server | EC2 | Compute Engine (GCE) | | Virtual Machine | DigitalOcean | Oper |
| PaaS | Elastic Beanstalk | App Engine | App Engine | Web Apps | Heroku, AppFog, OpenShift | Mete AppS Clou Foun Conv |
| Serverless, microservices | Lambda, API Gateway | Functions | | Function Apps | PubNub Blocks, Auth0 Webtask | Kong |
| Container, cluster | ECS, EKS, Fargate | Container Engine, | Borg or Omega | Container Service | | Kube Meso |

| Service | AWS | Google Cloud | Google Internal | Microsoft Azure | Other providers | Ope "bu |
|---|---|---|---|---|---|---|
| manager | | Kubernetes | | | | |
| Object storage | S3 | Cloud Storage | GFS | Storage Account | DigitalOcean Spaces | Swift Mini |
| Block storage | EBS | Persistent Disk | | Storage Account | DigitalOcean Volumes | NFS |
| SQL datastore | RDS | Cloud SQL | | SQL Database | | MyS Post |
| Sharded RDBMS | | Cloud Spanner | F1, Spanner | Azure Database for PostgreSQL - Hyperscale (Citus) | | Crate Cock |
| Bigtable | | Cloud Bigtable | Bigtable | | | HBas |
| Key-value store, column store | DynamoDB | Cloud Datastore | Megastore | Tables, DocumentDB | | Cass Couc Reth Redi |
| Memory cache | ElastiCache | App Engine Memcache | | Redis Cache | | Mem Redi |
| Search | CloudSearch, Elasticsearch | | | Search | Algolia, QBox, Elastic Cloud | Elast Solr |

| Service | AWS | Google Cloud | Google Internal | Microsoft Azure | Other providers | Ope "bu |
|---|---|---|---|---|---|---|
| | (managed) | | | | | |
| Data warehouse | Redshift | BigQuery | Dremel | SQL Data Warehouse | Oracle, IBM, SAP, HP, many others | Gree |
| Business intelligence | QuickSight | Data Studio 360 | | Power BI | Tableau | |
| Lock manager | DynamoDB (weak) | | Chubby | Lease blobs in Storage Account | | Zook Etcd, |
| Message broker | SQS, SNS, IoT | Pub/Sub | PubSub2 | Service Bus | | Rabb Kafk |
| Streaming, distributed log | Kinesis | Dataflow | PubSub2 | Event Hubs | | Kafk Apex Spar Strea Storm |
| MapReduce | EMR | Dataproc | MapReduce | HDInsight, DataLake Analytics | Qubole | Hado |
| Monitoring | CloudWatch | Stackdriver Monitoring | Borgmon | Monitor | | Prom |
| Tracing | X-Ray | Stackdriver Trace | | Monitor (Application | DataDog, New Relic, Epsagon | Zipki Appo |

| Service | AWS | Google Cloud | Google Internal | Microsoft Azure | Other providers | Ope "bu |
|---|---|---|---|---|---|---|
| | | | | Insights) | | |
| Metric management | | | Borgmon, TSDB | Application Insights | | Grap Influ Oper Grafa Riem Prom |
| CDN | CloudFront | Cloud CDN | | CDN | Akamai, Fastly, Cloudflare, Limelight Networks | Apac Serv |
| Load balancer | CLB/ALB | Load Balancing | GFE | Load Balancer, Application Gateway | | ngin HAP Apac Serv |
| DNS | Route53 | DNS | | DNS | | bind |
| Email | SES | | | | Sendgrid, Mandrill, Postmark | |
| Git hosting | CodeCommit | Cloud Source Repositories | | Visual Studio Team Services | GitHub, BitBucket | GitLa |
| User authentication | Cognito | Firebase Authentication | | Azure Active Directory | | oaut |

| Service | AWS | Google Cloud | Google Internal | Microsoft Azure | Other providers | Ope "bu |
|---|---|---|---|---|---|---|
| Mobile app analytics | Mobile Analytics | Firebase Analytics | | HockeyApp | Mixpanel | |
| Mobile app testing | Device Farm | Firebase Test Lab | | Xamarin Test Cloud | BrowserStack, Sauce Labs, Testdroid | |
| Managing SSL/TLS certificates | Certificate Manager | | | | Let's Encrypt, Comodo, Symantec, GlobalSign | |
| Automatic speech recognition and natural language understanding | Transcribe (ASR), Lex (NLU) | Cloud Speech API, Natural Language API | | Cognitive services | AYLIEN Text Analysis API, Ambiverse Natural Language Understanding API | Stan Core Suite Oper Apac spaC |
| Text-to-speech engine in the cloud | Polly | | | | Nuance, Vocalware, IBM | Mim Mary |
| Image recognition | Rekognition | Vision API | | Cognitive services | IBM Watson, Clarifai | Tens Oper |
| OCR (Text recognition) | Textract (documents), | Cloud Vision API | | Computer Vision API | | Tesse |

| Service | AWS | Google Cloud | Google Internal | Microsoft Azure | Other providers | Ope "bu |
|---|---|---|---|---|---|---|
| | Rekognition (photographs) | | | | | |
| Language Translation | Translate | Translate | | Translator Text API | | Aper |
| File Share and Sync | WorkDocs | Google Docs | | OneDrive | Dropbox, Box, Citrix File Share | own |
| Machine Learning | SageMaker, DeepLens, ML | ML Engine, Auto ML | | ML Studio | Watson ML | |
| Data Loss Prevention | Macie | Cloud Data Loss Prevention | | Azure Information Protection | | |

🚧 *Please help fill this table in.*

Selected resources with more detail on this chart:

- Google internal: MapReduce, Bigtable, Spanner, F1 vs Spanner, Bigtable vs Megastore

Back to top ⬆️

## AWS Product Maturity and Releases

It's important to know the maturity of each AWS product. Here is a mostly complete list of first release date, with link
Most recently released services are first. Not all services are available in all regions; see this table.

| Service | Original release | Availability | CLI Support | HIPAA Compliant |
|---|---|---|---|---|
| 🐥 X-Ray | 2016-12 | General | ✓ | ✓ |

| Service | Original release | Availability | CLI Support | HIPAA Compliant |
|---|---|---|---|---|
| Lex | 2016-11 | Preview | | |
| Polly | 2016-11 | General | ✓ | ✓ |
| Rekognition | 2016-11 | General | ✓ | ✓ |
| Athena | 2016-11 | General | ✓ | ✓ |
| Batch | 2016-11 | General | ✓ | ✓ |
| Database Migration Service | 2016-03 | General | | ✓ |
| Certificate Manager | 2016-01 | General | ✓ | ✓ |
| IoT | 2015-08 | General | ✓ | ✓ |
| WAF | 2015-10 | General | ✓ | ✓ |
| Data Pipeline | 2015-10 | General | ✓ | |
| Elasticsearch | 2015-10 | General | ✓ | ✓ |
| Aurora | 2015-07 | General | ✓ | ✓[3] |
| Service Catalog | 2015-07 | General | ✓ | ✓ |
| Device Farm | 2015-07 | General | ✓ | |
| CodePipeline | 2015-07 | General | ✓ | ✓ |
| CodeCommit | 2015-07 | General | ✓ | ✓ |
| API Gateway | 2015-07 | General | ✓ | ✓[1] |
| Config | 2015-06 | General | ✓ | ✓ |
| EFS | 2015-05 | General | ✓ | ✓ |

| Service | Original release | Availability | CLI Support | HIPAA Compliant |
|---|---|---|---|---|
| Machine Learning | 2015-04 | General | ✓ | |
| Lambda | 2014-11 | General | ✓ | ✓ |
| ECS | 2014-11 | General | ✓ | ✓ |
| EKS | 2018-06 | General | ✓[12] | ✓ |
| KMS | 2014-11 | General | ✓ | ✓ |
| CodeDeploy | 2014-11 | General | ✓ | ✓ |
| Kinesis | 2013-12 | General | ✓ | ✓ |
| CloudTrail | 2013-11 | General | ✓ | ✓ |
| AppStream | 2013-11 | Preview | | ✓ |
| CloudHSM | 2013-03 | General | ✓ | ✓ |
| Silk | 2013-03 | Obsolete? | | |
| OpsWorks | 2013-02 | General | ✓ | ✓ |
| Redshift | 2013-02 | General | ✓ | ✓ |
| Elastic Transcoder | 2013-01 | General | ✓ | |
| Glacier | 2012-08 | General | ✓ | ✓ |
| CloudSearch | 2012-04 | General | ✓ | |
| SWF | 2012-02 | General | ✓ | ✓ |
| Storage Gateway | 2012-01 | General | ✓ | ✓ |
| DynamoDB | 2012-01 | General | ✓ | ✓ |

| Service | Original release | Availability | CLI Support | HIPAA Compliant |
|---|---|---|---|---|
| DirectConnect | 2011-08 | General | ✓ | ✓ |
| ElastiCache | 2011-08 | General | ✓ | ✓[14] |
| CloudFormation | 2011-04 | General | ✓ | ✓ |
| SES | 2011-01 | General | ✓ | ✓ |
| Elastic Beanstalk | 2010-12 | General | ✓ | ✓ |
| Route 53 | 2010-10 | General | ✓ | ✓ |
| IAM | 2010-09 | General | ✓ | |
| SNS | 2010-04 | General | ✓ | ✓ |
| EMR | 2010-04 | General | ✓ | ✓ |
| RDS | 2009-12 | General | ✓ | ✓[2] |
| VPC | 2009-08 | General | ✓ | ✓ |
| Snowball | 2015-10 | General | ✓ | ✓ |
| Snowmobile | 2016-11 | General | | ✓ |
| CloudWatch | 2009-05 | General | ✓ | ✓ |
| CloudFront | 2008-11 | General | ✓ | ✓[4] |
| Fulfillment Web Service | 2008-03 | Obsolete? | | |
| SimpleDB | 2007-12 | ! Nearly obsolete | ✓ | |
| DevPay | 2007-12 | General | | |
| Flexible Payments Service | 2007-08 | Retired | | |

| Service | Original release | Availability | CLI Support | HIPAA Compliant |
|---|---|---|---|---|
| EC2 | 2006-08 | General | ✓ | ✓[5,6,7] |
| SQS | 2006-07 | General | ✓ | ✓ |
| S3 | 2006-03 | General | ✓ | ✓[8] |
| Alexa Top Sites | 2006-01 | General ❗ HTTP-only | | |
| Alexa Web Information Service | 2005-10 | General ❗ HTTP-only | | |

Back to top ⬆️

**Footnotes**

1: Excludes use of Amazon API Gateway caching
2: RDS MySQL, Oracle, and PostgreSQL engines only
3: MySQL-compatible Aurora edition only
4: Excludes Lambda@Edge
5: Includes EC2 Systems Manager
6: Includes Elastic Block Storage (EBS)
7: Includes Elastic Load Balancing
8: Includes S3 Transfer Acceleration
9: Includes RDS MySQL, Oracle, PostgreSQL, SQL Server, and MariaDB
10: Includes Auto-Scaling
11: Data Analytics, Streams, Video Streams and Firehose
12: Kubernetes uses a custom CLI for Pod/Service management called kubectl. AWS CLI only handles Kubernetes Mas
13: IoT Core (includes Device Management) and Greengrass
14: ElastiCache for Redis only
15: Snowball and Snowball Edge

## Compliance

- Many applications have strict requirements around reliability, security, or data privacy. The AWS Compliance pag certifications, which include **PCI DSS Level 1**, **SOC 1,2, and 3**, **HIPAA**, and **ISO 9001**.
- Security in the cloud is a complex topic, based on a shared responsibility model, where some elements of compl AWS, and some are provided by your company.
- Several third-party vendors offer assistance with compliance, security, and auditing on AWS. If you have substan assistance is a good idea.
- From inside **China**, AWS services outside China are generally accessible, though there are at times breakages in s services inside China.

## Getting Help and Support

- **Forums:** For many problems, it's worth searching or asking for help in the discussion forums to see if it's a know
- **Premium support:** AWS offers several levels of premium support.
  - The first tier, called "Developer support" lets you file support tickets with 12 to 24 hour turnaround time, it s monthly spend reaches around $1000 it changes to a 3% surcharge on your bill.
  - The higher-level support services are quite expensive — and increase your bill by up to 10%. Many large an never pay for this level of support. They are usually more helpful for midsize or larger companies needing ra or more perplexing problems.
  - Keep in mind, a flexible architecture can reduce need for support. You shouldn't be relying on AWS to solve example, if you can easily re-provision a new server, it may not be urgent to solve a rare kernel-level issue u If your EBS volumes have recent snapshots, you may be able to restore a volume before support can rectify volume. If your services have an issue in one availability zone, you should in any case be able to rely on a re services to another zone.
  - Larger customers also get access to AWS Enterprise support, with dedicated technical account managers (TA time SLAs.
  - There is definitely some controversy about how useful the paid support is. The support staff don't always se information and authority to solve the problems that are brought to their attention. Often your ability to ha depend on your relationship with your account rep.
- **Account manager:** If you are at significant levels of spend (thousands of US dollars plus per month), you may be ask for) a dedicated account manager.

    - These are a great resource, even if you're not paying for premium support. Build a good relationship with th for questions, problems, and guidance.
    - Assign a single point of contact on your company's side, to avoid confusing or overwhelming them.
- **Contact:** The main web contact point for AWS is here. Many technical requests can be made via these channels.
- **Consulting and managed services:** For more hands-on assistance, AWS has established relationships with many managed service partners. The big consultants won't be cheap, but depending on your needs, may save you cos you set up your architecture more effectively, or offering specific expertise, e.g. security. Managed service provic full-service management of cloud resources.
- **AWS Professional Services:** AWS provides consulting services alone or in combination with partners.

### Restrictions and Other Notes

- 🔸 Lots of resources in Amazon have **limits** on them. This is actually helpful, so you don't incur large costs accide that quotas be increased by opening support tickets. Some limits are easy to raise, and some are not. (Some of t below.) Additionally, not all service limits are published.
    - **Obtaining Current Limits and Usage:** Limit information for a service may be available from the service API, neither (in which case you'll need to contact Support). This page from the awslimitchecker tool's documenta summary of available retrieval options for each limit. The tool itself is also valuable for automating limit che
- 🔸 **AWS terms of service** are extensive. Much is expected boilerplate, but it does contain important notes and re In particular, there are restrictions against using many AWS services in **safety-critical systems**. (Those appreciativ to review clause 42.10.)

### Related Topics

- OpenStack is a private cloud alternative to AWS used by large companies that wish to avoid public cloud offerin

## Learning and Career Development

### Certifications

- **Certifications:** AWS offers **certifications** for IT professionals who want to demonstrate their knowledge.

- Certified Cloud Practitioner
- Certified Solutions Architect Associate
- Certified Developer Associate
- Certified SysOps Administrator Associate
- Certified Solutions Architect Professional
- Certified DevOps Engineer Professional
- Certified Security – Specialty
- Certified Advanced Networking – Specialty
- Certified Machine Learning – Specialty
- Certified Data Analytics – Specialty
- Certified Database – Specialty

Associate level certifications were once required as pre-requisites to taking the Professional examinations - this is no

- **Getting certified:** If you're interested in studying for and getting certifications, this practical overview tells you a know. The official page is here and there is an FAQ.
- **Training for certifications:** Training is offered by AWS themselves (mainly instructor-led and on-site) and various (usually as video-based training) such as A Cloud Guru, CloudAcademy and Linux Academy.
- **Do you need a certification?** Especially in consulting companies or when working in key tech roles in large non- certifications are important credentials. In others, including in many tech companies and startups, certifications a considered necessary. (In fact, fairly or not, some Silicon Valley hiring managers and engineers see them as a "ne resume.)

Certifications are required to access certificate lounges at official AWS events such as Summits and re:Invent. Lounge charging points, seats and relatively better coffee.

## Managing AWS

### Managing Infrastructure State and Change

A great challenge in using AWS to build complex systems (and with DevOps in general) is to manage infrastructure s
In general, this boils down to three broad goals for the state of your infrastructure:

- **Visibility**: Do you know the state of your infrastructure (what services you are using, and exactly how)? Do you a anyone on your team — make changes? Can you detect misconfigurations, problems, and incidents with your se
- **Automation**: Can you reconfigure your infrastructure to reproduce past configurations or scale up existing ones manual work, or requiring knowledge that's only in someone's head? Can you respond to incidents easily or aut
- **Flexibility**: Can you improve your configurations and scale up in new ways without significant effort? Can you ad the same tools? Do you share, review, and improve your configurations within your team?

Much of what we discuss below is really about how to improve the answers to these questions.

There are several approaches to deploying infrastructure with AWS, from the console to complex automation tools, t which attempt to help achieve visibility, automation, and flexibility.

## AWS Configuration Management

The first way most people experiment with AWS is via its web interface, the AWS Console. But using the Console is a often works against automation or flexibility.

So if you're not going to manage your AWS configurations manually, what should you do? Sadly, there are no simple approach has pros and cons, and the approaches taken by different companies vary widely, and include directly usin tooling on top yourself), using command-line tools, and using third-party tools and services.

## AWS Console

- The AWS Console lets you control much (but not all) functionality of AWS via a web interface.
- Ideally, you should only use the AWS Console in a few specific situations:
  - It's great for read-only usage. If you're trying to understand the state of your system, logging in and browsi
  - It is also reasonably workable for very small systems and teams (for example, one engineer setting up one s often).

    - It can be useful for operations you're only going to do rarely, like less than once a month (for example, a on probably won't revisit for a year). In this case using the console can be the simplest approach.
- ❗ **Think before you use the console:** The AWS Console is convenient, but also the enemy of automation, repro communication. If you're likely to be making the same change multiple times, avoid the console. Favor some sor have a path toward automation, as discussed next. Not only does using the console preclude automation, which prevents documentation, clarity, and standardization around processes for yourself and your team.

## Command-Line tools

- The aws command-line interface (CLI), used via the **aws** command, is the most basic way to save and automate
- Don't underestimate its power. It also has the advantage of being well-maintained — it covers a large proportio is up to date.
- In general, whenever you can, prefer the command line to the AWS Console for performing operations.
- 🔹 Even in the absence of fancier tools, you can **write simple Bash scripts** that invoke *aws* with specific argumer Git. This is a primitive but effective way to document operations you've performed. It improves automation, allow on a team, and gives others a starting point for future work.
- 🔹 For use that is primarily interactive (not scripted), consider instead using the **aws-shell** tool from AWS. It is ea completion and a colorful UI, but still works on the command line. If you're using SAWS, a previous version of th migrate to aws-shell.

## APIs and SDKs

- **SDKs** for using AWS APIs are available in most major languages, with Go, iOS, Java, JavaScript, Python, Ruby, and used. AWS maintains a short list, but the awesome-aws list is the most comprehensive and current. Note suppor
- **Retry logic:** An important aspect to consider whenever using SDKs is error handling; under heavy use, a wide va programming errors to throttling to AWS-related outages or failures, can be expected to occur. SDKs typically in backoff to address this, but this may need to be understood and adjusted over time for some applications. For e to alert on some error codes and not on others.
- ❗ Don't use APIs directly. Although AWS documentation includes lots of API details, it's better to use the SDKs language to access APIs. SDKs are more mature, robust, and well-maintained than something you'd write yourse

## Boto

- A good way to automate operations in a custom way is **Boto3**, also known as the Amazon SDK for Python. **Boto** this library, has been in wide use for years, but now there is a newer version with official support from Amazon, s projects.
- Boto3 contains a variety of APIs that operate at either a high level or a low level, here some explanation of both:
  - The low level APIs (Client APIs) are mapped to AWS Cloud service-specific APIs, and all service operations ar Clients are generated from a JSON service definition file.
  - The high level option, Resource APIs, allows you to avoid calling the network at the low level and instead pr way to interact with AWS Cloud services.
- Boto3 has a lot of helpful **features** like *waiters*, which provide a structure that allows for code to wait for change example, when you are creating an EC2 instance and need wait until the instance is running in order to perform
- If you find yourself writing a Bash script with more than one or two CLI commands, you're probably doing it wro writing a Boto script instead. This has the advantages that you can:
  - Check return codes easily so success of each step depends on success of past steps.
  - Grab interesting bits of data from responses, like instance ids or DNS names.
  - Add useful environment information (for example, tag your instances with git revisions, or inject the latest b initialization script).

Back to top ⬆️

## General Visibility

- 🔹 **Tagging resources** is an essential practice, especially as organizations grow, to better understand your resour through automation or convention, you can add tags:
  - For the org or developer that "owns" that resource
  - For the product that resource supports
  - To label lifecycles, such as temporary resources or one that should be deprovisioned in the future
  - To distinguish production-critical infrastructure (e.g. serving systems vs backend pipelines)
  - To distinguish resources with special security or compliance requirements

- To (once enabled) allocate cost. Note that cost allocation tags only apply on a forward-looking basis; you ca
  them to items already billed.
- For many years, there was a notorious 10 tag limit per resource, which could not be raised and caused many
  pain. As of 2016, this was raised to 50 tags per resource.
- 🔹 In 2017, AWS introduced the ability to enforce tagging on instance and volume creation, deprecating po
  such as Cloud Custodian.
- 🔸 Tags are case sensitive; 'environment' and 'Environment' are two different tags. Automation in setting ta
  sensible option at significant scale.
- 🔸 There is a bug in the ASG console where spaces after tag names are preserved. So if you type "Name " w
  will not get the expected behavior. This is probably true in other locations and SDKs also. Be sure you do no
  keys unless you really mean it. (As of Jul 2018)
- 🔸 When resources are shared across the org, tags are not shared with it. For example, sharing Transit Gate
  correct tags in the account that created these resources but not in the accounts where these resources were

## Managing Servers and Applications

Back to top ⬆️

### AWS vs Server Configuration

This guide is about AWS, not DevOps or server configuration management in general. But before getting into AWS in
that in addition to the configuration management for your AWS resources, there is the long-standing problem of con
for servers themselves.

Back to top ⬆️

### Philosophy

- Heroku's Twelve-Factor App principles list some established general best practices for deploying applications.
- **Pets vs cattle:** Treat servers like cattle, not pets. That is, design systems so infrastructure is disposable. It should l
  server is unexpectedly destroyed.

- The concept of **immutable infrastructure** is an extension of this idea.
- Minimize application state on EC2 instances. In general, instances should be able to be killed or die unexpectedl State that is in your application should quickly move to RDS, S3, DynamoDB, EFS, or other data stores not on tha option, though it generally should not be the bootable volume, and EBS will require manual or automated re-m

Back to top ⬆️

## Server Configuration Management

- There is a large set of open source tools for managing configuration of server instances.
- These are generally not dependent on any particular cloud infrastructure, and work with any variety of Linux (or operating systems).
- Leading configuration management tools are Puppet, Chef, Ansible, and Saltstack. These aren't the focus of this mention them as they relate to AWS.

Back to top ⬆️

## Containers and AWS

- Docker and the containerization trend are changing the way many servers and services are deployed in general.
- Containers are designed as a way to package up your application(s) and all of their dependencies in a known wa container, you are including every library or binary your application needs, outside of the kernel. A big advantag it's easy to test and validate a container locally without worrying about some difference between your computer deploy on.
- A consequence of this is that you need fewer AMIs and boot scripts; for most deployments, the only boot script that fetches an exported docker image and runs it.
- Companies that are embracing microservice architectures will often turn to container-based deployments.
- AWS launched ECS as a service to manage clusters via Docker in late 2014, though many people still deploy Doc See the ECS section for more details.
- AWS launched EKS as a service to manage Kubernetes Clusters mid 2018, though many people still deploy ECS o themselves. See the EKS section for more details.

Back to top ⬆️

## Visibility

- Store and track instance metadata (such as instance id, availability zone, etc.) and deployment info (application b
  your logs or reports. The **instance metadata service** can help collect some of the AWS data you'll need.
- **Use log management services:** Be sure to set up a way to view and manage logs externally from servers.
  - Cloud-based services such as Sumo Logic, Splunk Cloud, Scalyr, LogDNA, and Loggly are the easiest to set u
    most expensive, which may be a factor depending on how much log data you have).
  - Major open source alternatives include Elasticsearch, Logstash, and Kibana (the "Elastic Stack") and Graylog.
  - If you can afford it (you have little data or lots of money) and don't have special needs, it makes sense to us
    whenever possible, since setting up your own scalable log processing systems is notoriously time consumin
- **Track and graph metrics:** The AWS Console can show you simple graphs from CloudWatch, you typically will wa
  kinds of metrics, from CloudWatch and your applications. Collect and export helpful metrics everywhere you can
  manageable enough you can afford it).
  - Services like Librato, KeenIO, and Datadog have fancier features or better user interfaces that can save a lot
    comparison is here.)
  - Use Prometheus or Graphite as timeseries databases for your metrics (both are open source).
  - Grafana can visualize with dashboards the stored metrics of both timeseries databases (also open source).

Back to top ⬆️

## Tips for Managing Servers

- ❗ **Timezone settings on servers**: unless *absolutely necessary*, always **set the timezone on servers to UTC** (see i
  distribution, such as Ubuntu, CentOS or Amazon Linux). Numerous distributed systems rely on time for synchron
  and UTC provides the universal reference plane: it is not subject to daylight savings changes and adjustments in
  you a lot of headache debugging elusive timezone issues and provide coherent timeline of events in your loggi
- **NTP and accurate time:** If you are not using Amazon Linux (which comes preconfigured), you should confirm yo
  correctly, to avoid insidious time drift (which can then cause all sorts of issues, from breaking API calls to mislea

part of your automatic configuration for every server. If time has already drifted substantially (generally >1000 s won't shift it back, so you may need to remediate manually (for example, like this on Ubuntu).

- **Testing immutable infrastructure:** If you want to be proactive about testing your service's ability to cope with in failure, it can be helpful to introduce random instance termination during business hours, which will expose any engineers are available to identify and fix them. Netflix's Simian Army (specifically, Chaos Monkey) is a popular t chaos-lambda by the BBC is a lightweight option which runs on AWS Lambda.

## Security and IAM

We cover security basics first, since configuring user accounts is something you usually have to do early on when set

### Security and IAM Basics

- 📒 IAM Homepage · User guide · FAQ
- The AWS Security Blog is one of the best sources of news and information on AWS security.
- **IAM** is the service you use to manage accounts and permissioning for AWS.
- Managing security and access control with AWS is critical, so every AWS administrator needs to use and underst level.
- IAM identities include users (people or services that are using AWS), groups (containers for sets of users and the (containers for permissions assigned to AWS service instances). Permissions for these identities are governed by pre-defined policies or custom policies that you create.
- IAM manages various kinds of authentication, for both users and for software services that may need to authent
  - **Passwords** to log into the console. These are a username and password for real users.
  - **Access keys**, which you may use with command-line tools. These are two strings, one the "id", which is an u of the form 'AXXXXXXXXXXXXXXXXXXX', and the other is the secret, which is a 40-character mixed-case bas often set up for services, not just users.
    - 📜 Access keys that start with AKIA are normal keys. Access keys that start with ASIA are session/tempo will require an additional "SessionToken" parameter to be sent along with the id and secret. See the do complete list of access key prefixes.

    - Multi-factor authentication (MFA), which is the highly recommended practice of using a keychain fob or sm layer of protection for user authentication.
- IAM allows complex and fine-grained control of permissions, dividing users into groups, assigning permissions t a policy language that can be used to customize security policies in a fine-grained way.
    - An excellent high level overview of IAM policy concepts lives at IAM Policies In A Nutshell.
    - 🔸 The policy language has a complex and error-prone JSON syntax that's quite confusing, so unless you ar base yours off trusted examples or AWS' own pre-defined managed policies.
- At the beginning, IAM policy may be very simple, but for large systems, it will grow in complexity, and need to b
    - 🔹 Make sure one person (perhaps with a backup) in your organization is formally assigned ownership of m make sure every administrator works with that person to have changes reviewed. This goes a long way to av serious misconfigurations.
- It is best to give each user or service the minimum privileges needed to perform their duties. This is the principle the foundations of good security. Organize all IAM users and groups according to levels of access they need.
- IAM has the permission hierarchy of:
    i. Explicit deny: The most restrictive policy wins.
    ii. Explicit allow: Access permissions to any resource has to be explicitly given.
    iii. Implicit deny: All permissions are implicitly denied by default.
- You can test policy permissions via the AWS IAM policy simulator tool. This is particularly useful if you write cust

Back to top ⬆️

## Security and IAM Tips

- 🔹 Use IAM to create individual user accounts and **use IAM accounts for all users from the beginning**. This is sl that much.
    - That way, you define different users, and groups with different levels of privilege (if you want, choose from A suggestions, of administrator, power user, etc.).
    - This allows credential revocation, which is critical in some situations. If an employee leaves, or a key is comp credentials with little effort.
    - You can set up Active Directory federation to use organizational accounts in AWS.

- ❗ **Enable MFA** on your account.
  - You should always use MFA, and the sooner the better — enabling it when you already have many users is e
  - Unfortunately it can't be enforced in software, so an administrative policy has to be established.
  - Most users can use the Google Authenticator app (on iOS or Android) to support two-factor authentication consider a hardware fob.
- ❗ Restrict use of significant IAM credentials as much as possible. Remember that in the cloud, loss of a highly c could essentially mean "game over," for your deployment, your users, or your whole company.
  - **Do NOT use the Root User account** other than when you initially create your account. Create custom IAM u those for your applications instead.
    - Lock up access and use of the root credentials as much as possible. Ideally they should be effectively "c deployments, this means attached to an actual MFA device, physically secured and rarely used.
- ❗ **Turn on CloudTrail:** One of the first things you should do is enable CloudTrail. Even if you are not a security h not to do this from the beginning, so you have data on what has been happening in your AWS account should y You'll likely also want to set up a log management service to search and access these logs.
- 🔹 **Use IAM roles for EC2:** Rather than assign IAM users to applications like services and then sharing the sensiti assign roles to EC2 instances and have applications retrieve credentials from the instance metadata.
- Assign IAM roles by realm — for example, to development, staging, and production. If you're setting up a role, i specific realm so you have clean separation. This prevents, for example, a development instance from connectin
- **Best practices:** AWS' list of best practices is worth reading in full up front.
- **IAM Reference:** This interactive reference for all IAM actions, effects, and resources is great to have open while v understand existing IAM policies.
- **Multiple accounts:** Decide on whether you want to use multiple AWS accounts and research how to organize ac to consider:
  - Number of users
  - Importance of isolation
    - Resource Limits
    - Permission granularity
    - Security
    - API Limits

    - Regulatory issues
    - Workload
    - Size of infrastructure
    - Cost of multi-account "overhead": Internal AWS service management tools may need to be custom built or
    - 🔹 It can help to use separate AWS accounts for independent parts of your infrastructure if you expect a hig since AWS throttles calls at the AWS account level.
- Inspector is an automated security assessment service from AWS that helps identify common security risks. This adhere to certain security practices and may help with compliance.
- Trusted Advisor addresses a variety of best practices, but also offers some basic security checks around IAM usa configurations, and MFA. At paid support tiers, Trusted Advisor exposes additional checks around other areas, su optimization.
- **Use KMS for managing keys**: AWS offers KMS for securely managing encryption keys, which is usually a far bett key security yourself. See below.
- AWS WAF is a web application firewall to help you protect your applications from common attack patterns.
- **Security auditing:**
    - Security Monkey is an open source tool that is designed to assist with security audits.
    - Scout2 is an open source tool that uses AWS APIs to assess an environment's security posture. Scout2 is sta maintained.
    - 🔹 **Export and audit security settings:** You can audit security policies simply by exporting settings using AW script like SecConfig.py (from this 2013 talk) and then reviewing and monitoring changes manually or autor

Back to top ⬆️

## Security and IAM Gotchas and Limitations

- ❗ **Don't share user credentials:** It's remarkably common for first-time AWS users to create one account and on key or password), and then use them for a while, sharing among engineers and others within a company. This is is an insecure practice for many reasons, but in particular, if you do, you will have reduced ability to revoke cred per-service basis (for example, if an employee leaves or a key is compromised), which can lead to serious compl

- ❗ **Instance metadata throttling:** The instance metadata service has rate limiting on API calls. If you deploy IAM
should!) and have lots of services, you may hit global account limits easily.
    - One solution is to have code or scripts cache and reuse the credentials locally for a short period (say 2 minu
be put into the ~/.aws/credentials file but must also be refreshed automatically.
    - But be careful not to cache credentials for too long, as they expire. (Note the other dynamic metadata also
should not be cached a long time, either.)
- 🔸 Some IAM operations are slower than other API calls (many seconds), since AWS needs to propagate these g
- ❗ The uptime of IAM's API has historically been lower than that of the instance metadata API. Be wary of incorp
IAM's API into critical paths or subsystems — for example, if you validate a user's IAM group membership when
and aren't careful about precaching group membership or maintaining a back door, you might end up locking u
the API isn't available.
- ❗ **Don't check in AWS credentials or secrets to a git repository.** There are bots that scan GitHub looking for cr
tools, such as git-secrets to prevent anyone on your team from checking in sensitive information to your git repo

## S3

### S3 Basics

- 📒 Homepage · Developer guide · FAQ · Pricing
- **S3** (Simple Storage Service) is AWS' standard cloud storage service, offering file (opaque "blob") storage of arbit
almost any size, from 0 to **5TB**. (Prior to 2011 the maximum size was 5 GB; larger sizes are now well supported v
- Items, or **objects**, are placed into named **buckets** stored with names which are usually called **keys**. The main con
- Objects are created, deleted, or updated. Large objects can be streamed, but you cannot modify parts of a value
whole object. Partial data access can work via S3 Select.
- Every object also has **metadata**, which includes arbitrary key-value pairs, and is used in a way similar to HTTP he
system-defined, some are significant when serving HTTP content from buckets or CloudFront, and you can also
for your own use.
- **S3 URIs:** Although often bucket and key names are provided in APIs individually, it's also common practice to w
form 's3://bucket-name/path/to/key' (where the key here is 'path/to/key'). (You'll also see 's3n://' and 's3a://' pre

- **S3 vs Glacier, EBS, and EFS:** AWS offers many storage services, and several besides S3 offer file-type abstractions infrequently accessed archival storage. EBS, unlike S3, allows random access to file contents via a traditional files attached to one EC2 instance at a time. EFS is a network filesystem many instances can connect to, but at higher table.

Back to top ⬆️

## S3 Tips

- For most practical purposes, you can consider S3 capacity unlimited, both in total size of files and number of obj objects in a bucket is essentially also unlimited. Customers routinely have millions of objects.
- ❗ **Permissions:**
  - 🔸 If you're storing business data on Amazon S3, it's important to manage permissions sensibly. In 2017 cor and Verizon saw data breaches due to poorly-chosen S3 configuration for sensitive data. Fixing this later ca have a lot of assets and internal users.
  - 🔸 There are 3 different ways to grant permissions to access Amazon S3 content in your buckets.
    - **IAM policies** use the familiar Identity and Access Management permission scheme to control access to
    - **Bucket policies** grant or deny permissions to an entire bucket. You might use this when hosting a webs bucket publicly readable, or to restrict access to a bucket by IP address. Amazon's sample bucket polici cases where these policies come in handy.
    - **Access Control Lists** (ACLs) can also be applied to every bucket and object stored in S3. ACLs grant add beyond those specified in IAM or bucket policies. ACLs can be used to grant access to another AWS use like the general public. This is powerful but can be dangerous, because you need to inspect every objec
  - 🔸 AWS' predefined access control groups allow access that may not be what you'd expect from their name
    - **"All Users", or "Everyone", grants permission to the general public**, not only to users defined in your object is available to All Users, then it can be retrieved with a simple HTTP request of the form `http://name/filename`. No authorization or signature is required to access data in this category.
    - **"Authenticated Users" grants permissions to anyone with an AWS account**, again not limited to your anyone can sign up for AWS, for all intents and purposes **this is also open to the general public**.
    - **"Log Delivery" group is used by AWS to write logs to buckets** and should be safe to enable on the bu

        - A typical use case of this ACL is used in conjunction with the requester pays functionality of S3.
    - ❗ Bucket permissions and object permissions are two different things and independent of each other. A p bucket can be seen when listing the bucket, but not downloaded. At the same time, a public object in a priv because the bucket contents can't be listed, but can still be downloaded by anyone who knows its exact key access to set bucket permissions can still make objects public if they have `s3:PutObjectAcl` or `s3:PutObjec`
    - 🐥 In August 2017, AWS added AWS Config rules to ensure your S3 buckets are secure.
        - ❗ These AWS Config rules only check the security of your bucket policy and bucket-level ACLs. You ca that grant additional permissions, including opening files to the whole world.
    - 🔹 Do create new buckets if you have different types of data with different sensitivity levels. This is much les complex permissions rules. For example, if data is for administrators only, like log data, put it in a new bucke can access.
    - For more guidance, see:
        - How to Secure an Amazon S3 Bucket
        - Deep dive into S3 access controls.
        - How do S3 permissions work?.
- **Bucket naming:** Buckets are chosen from a global namespace (across all regions, even though S3 itself stores da you select), so you'll find many bucket names are already taken. Creating a bucket means taking ownership of th Bucket names have a few restrictions on them.
    - Bucket names can be used as part of the hostname when accessing the bucket or its contents, like `<bucket_ 1.amazonaws.com`, as long as the name is DNS compliant.
    - A common practice is to use the company name acronym or abbreviation to prefix (or suffix, if you prefer D bucket names (but please, don't use a check on this as a security measure — this is highly insecure and easi
    - 🔸 Bucket names with '.' (periods) in them can cause certificate mismatches when used with SSL. Use '-' inst conforms with both SSL expectations and is DNS compliant.
- **Versioning:** S3 has optional versioning support, so that all versions of objects are preserved on a bucket. This is an archive of changes or the ability to back out mistakes (caution: it lacks the featureset of full version control sy
- **Durability:** Durability of S3 is extremely high, since internally it keeps several replicas. If you don't delete it by ac S3 not losing your data. (AWS offers the seemingly improbable durability rate of 99.999999999%, but this is a m

based on independent failure rates and levels of replication — not a true probability estimate. Either way, S3 has durability.) Note this is *much* higher durability than EBS!

- 💸 **S3 pricing** depends on storage, requests, and transfer.
  - For transfer, putting data into AWS is free, but you'll pay on the way out. Transfer from S3 to EC2 in the *sam* other regions or the Internet in general is not free.
  - Deletes are free.
- **S3 Reduced Redundancy and Infrequent Access:** Most people use the Standard storage class in S3, but there ar with lower cost:
  - 🔸 Reduced Redundancy Storage (RRS) has been effectively deprecated, and has lower durability (99.99%, s standard S3. Note that it no longer participates in S3 price reductions, so it offers worse redundancy for mo S3. As a result, there's no reason to use it.
  - Infrequent Access (IA) lets you get cheaper storage in exchange for more expensive access. This is great for already processed, but might want to look at later. To get an idea of the cost savings when using Infrequent this S3 Infrequent Access Calculator.
  - S3 - Intelligent Tiering storage class is designed to optimize costs by automatically moving data to the mos without performance impact or operational overhead.
  - S3 - One Zone - IA is for data that is accessed less frequently, but requires rapid access when needed. Unlik which store data in a minimum of three Availability Zones (AZs), S3 One Zone-IA stores data in a single AZ Standard-IA.
  - Glacier is a third alternative discussed as a separate product.
  - See the comparison table.
- ⏱️ **Performance:** Maximizing S3 performance means improving overall throughput in terms of bandwidth and n second.
  - S3 is highly scalable, so in principle you can get arbitrarily high throughput. (A good example of this is S3Di
  - But usually you are constrained by the pipe between the source and S3 and/or the level of concurrency of c
  - Throughput is of course highest from within AWS to S3, and between EC2 instances and S3 buckets that are
  - Bandwidth from EC2 depends on instance type. See the "Network Performance" column at ec2instances.info
  - Throughput of many objects is extremely high when data is accessed in a distributed way, from many EC2 in read or write objects from S3 from hundreds or thousands of instances at once.

    - However, throughput is very limited when objects accessed sequentially from a single instance. Individual o milliseconds, and bandwidth to and from instances is limited.
    - Therefore, to perform large numbers of operations, it's necessary to use multiple worker threads and conne instances, and for larger jobs, multiple EC2 instances as well.
    - **Multi-part uploads:** For large objects you want to take advantage of the multi-part uploading capabilities (s chunk sizes of 5 MB).
    - **Large downloads:** Also you can download chunks of a single large object in parallel by exploiting the HTTP capability.
    - 🔸 **List pagination:** Listing contents happens at 1000 responses per request, so for buckets with many millio take time.
    - ❗ **Key prefixes:** Previously randomness in the beginning of key names was necessary in order to avoid hot necessary as of July, 2018.
    - For data outside AWS, DirectConnect and S3 Transfer Acceleration can help. For S3 Transfer Acceleration, y equivalent of 1-2 months of storage for the transfer in either direction for using nearer endpoints.
- **Command-line applications:** There are a few ways to use S3 from the command line:
    - Originally, s3cmd was the best tool for the job. It's still used heavily by many.
    - The regular aws command-line interface now supports S3 well, and is useful for most situations.
    - s4cmd is a replacement, with greater emphasis on performance via multi-threading, which is helpful for larg files, and also offers Unix-like globbing support.
- **GUI applications:** You may prefer a GUI, or wish to support GUI access for less technical users. Some options:
    - The AWS Console does offer a graphical way to use S3. Use caution telling non-technical people to use it, h permissions, it offers access to many other AWS features.
    - Transmit is a good option on macOS for most use cases.
    - Cyberduck is a good option on macOS and Windows with support for multipart uploads, ACLs, versioning, l storage classes and server side encryption (SSE-S3 and SSE-KMS).
- **S3 and CloudFront:** S3 is tightly integrated with the CloudFront CDN. See the CloudFront section for more infor transfer acceleration.
- **Static website hosting:**

    - S3 has a static website hosting option that is simply a setting that enables configurable HTTP index and erro
      support to public content in S3. It's a simple way to host static assets or a fully static website.
    - Consider using CloudFront in front of most or all assets:
        - Like any CDN, CloudFront improves performance significantly.
        - 🔸 SSL is only supported on the built-in amazonaws.com domain for S3. S3 supports serving these sites
          domain, but not over SSL on a custom domain. However, CloudFront allows you to serve a custom dom
          provides free SNI SSL/TLS certificates via Amazon Certificate Manager. SNI does not work on very outd
          systems. Alternatively, you can provide your own certificate to use on CloudFront to support all browse
          fee.
        - 🔸 If you are including resources across domains, such as fonts inside CSS files, you may need to confi
          serving those resources.
        - Since pretty much everything is moving to SSL nowadays, and you likely want control over the domain,
          up CloudFront with your own certificate in front of S3 (and to ignore the AWS example on this as it is n
        - That said, if you do, you'll need to think through invalidation or updates on CloudFront. You may wish t
          hashes in filenames so invalidation is not necessary.
- **Data lifecycles:**
    - When managing data, the understanding the lifecycle of the data is as important as understanding the data
      into a bucket, think about its lifecycle — its end of life, not just its beginning.
    - 🔹 In general, data with different expiration policies should be stored under separate prefixes at the top lev
      voluminous logs might need to be deleted automatically monthly, while other data is critical and should ne
      former in a separate bucket or at least a separate folder is wise.
    - 🔸 Thinking about this up front will save you pain. It's very hard to clean up large collections of files created
      varying lifecycles and no coherent organization.
    - Alternatively you can set a lifecycle policy to archive old data to Glacier. Be careful with archiving large num
      Glacier, since it may actually cost more.
    - There is also a storage class called **Infrequent Access** that has the same durability as Standard S3, but is dis
      suitable for objects that are infrequently accessed.
- **Data consistency:** Understanding data consistency is critical for any use of S3 where there are multiple producer

    - Creation and updates to individual objects in S3 are **atomic**, in that you'll never upload a new object or cha another client see only part half the change.
    - The uncertainty lies with *when* your clients and other clients see updates.
    - **New objects:** If you create a new object, you'll be able to read it instantly, which is called **read-after-write c**
        - Well, with the additional caveat that if you do a read on an object before it exists, then create it, you ge read-after-write).
        - This does not apply to any list operations; newly created objects are not guaranteed to appear in a list
    - **Updates to objects:** If you overwrite or delete an object, you're only guaranteed **eventual consistency**, i.e. t you have no guarantee of when.
    - 🔹 For many use cases, treating S3 objects as **immutable** (i.e. deciding by convention they will be created o can greatly simplify the code that uses them, avoiding complex state management.
    - 🔹 Note that until 2015, 'us-standard' region had had a weaker eventual consistency model, and the other ( after-write. This was finally corrected — but watch for many old blogs mentioning this!
    - **Slow updates:** In practice, "eventual consistency" usually means within seconds, but expect rare cases of mi
- **S3 as a filesystem:**
    - In general S3's APIs have inherent limitations that make S3 hard to use directly as a POSIX-style filesystem w own object format. For example, appending to a file requires rewriting, which cripples performance, and atc mutual exclusion on opening files, and hardlinks are impossible.
    - s3fs is a FUSE filesystem that goes ahead and tries anyway, but it has performance limitations and surprises
    - Riofs (C) and Goofys (Go) are more recent efforts that attempt adopt a different data storage format to add are likely improvements on s3fs.
    - S3QL (discussion) is a Python implementation that offers data de-duplication, snap-shotting, and encryptio time.
    - ObjectiveFS (discussion) is a commercial solution that supports filesystem features and concurrent clients.
- If you are primarily using a VPC, consider setting up a VPC Endpoint for S3 in order to allow your VPC-hosted re without the need for extra network configuration or hops.
- **Cross-region replication:** S3 has a feature for replicating a bucket between one region and another. Note that S replicated within one region, so usually this isn't necessary for durability, but it could be useful for compliance (g

data storage), lower latency, or as a strategy to reduce region-to-region bandwidth costs by mirroring heavily us region.

- **IPv4 vs IPv6:** For a long time S3 only supported IPv4 at the default endpoint `https://BUCKET.s3.amazonaws.com` . 2016 it now supports both IPv4 & IPv6! To use both, you have to enable dualstack either in your preferred API cl this url scheme `https://BUCKET.s3.dualstack.REGION.amazonaws.com` . This extends to S3 Transfer Acceleration as
- **S3 event notifications:** S3 can be configured to send an SNS notification, SQS message, or AWS Lambda functio
- 💸 Limit your individual users (or IAM roles) to the minimal required S3 locations, and catalog the "approved" lo tends to become the dumping ground where people put data to random locations that are not cleaned up for y bucks.
- If a bucket is deleted in S3, it can take up to 10 hours before a bucket with the same name can be created again

Back to top ⬆️

## S3 Gotchas and Limitations

- ❗ S3 buckets sit outside the VPC and can be accessed from anywhere in the world if bucket policies are not set permissions section above carefully, there are countless cases of buckets exposed to the public.
- 🔸 For many years, there was a notorious **100-bucket limit** per account, which could not be raised and caused m pain. As of 2015, you can request increases. You can ask to increase the limit, but it will still be capped (generally account).
- 🔸 Be careful not to make implicit assumptions about transactionality or sequencing of updates to objects. Neve modify a sequence of objects, the clients will see the same modifications in the same sequence, or if you upload that they will all appear at once to all clients.
- 🔸 S3 has an SLA with 99.9% uptime. If you use S3 heavily, you'll inevitably see occasional error accessing or sto infrastructure fail. Availability is usually restored in seconds or minutes. Although availability is not extremely hig durability is excellent.
- 🔸 After uploading, any change that you make to the object causes a full rewrite of the object, so avoid appendi regular files.
- 🔸 Eventual data consistency, as discussed above, can be surprising sometimes. If S3 suffers from internal replica be visible from a subset of the machines, depending on which S3 endpoint they hit. Those usually resolve within

seen isolated cases when the issue lingered for 20-30 hours.

- 🔸 **MD5s and multi-part uploads: I**n S3, the ETag header in S3 is a hash on the object. And in many cases, it is th is not the case in general when you use multi-part uploads. One workaround is to compute MD5s yourself and p header (such as is done by s4cmd).
- 🔸 **Incomplete multi-part upload costs: I**ncomplete multi-part uploads accrue storage charges even if the uploa created. Amazon (and others) recommend using a lifecycle policy to clean up incomplete uploads and save on s you have many of these, it may be worth investigating whatever's failing regularly.
- 🔸 **US Standard region:** Previously, the us-east-1 region (also known as the US Standard region) was replicated a greater variability of latency. Effective Jun 19, 2015 this is no longer the case. All Amazon S3 regions now suppor consistency. Amazon S3 also renamed the US Standard region to the US East (N. Virginia) region to be consisten naming conventions.
- 🔸 **S3 authentication versions and regions: I**n newer regions, S3 only supports the latest authentication. If an S3 SDK doesn't work in one region, but works correctly in another region, make sure you are using the latest authe

Back to top ⬆️

## Storage Durability, Availability, and Price

As an illustration of comparative features and price, the table below gives S3 Standard, RRS, IA, in comparison with G d2.xlarge instance store using **Virginia region** as of **Sept 2017**.

| | Durability (per year) | Availability "designed" | Availability SLA | Storage (per TB per month) | GET retrieve millic |
|---|---|---|---|---|---|
| **Glacier** | Eleven 9s | Sloooow | – | $4 | $50 |
| **S3 IA** | Eleven 9s | 99.9% | **99%** | $12.50 | $1 |
| ~~**S3 RRS**~~ | ~~**99.99%**~~ | ~~99.99%~~ | ~~99.9%~~ | ~~$24 (first TB)~~ | ~~$0.40~~ |
| **S3 Standard** | Eleven 9s | 99.99% | 99.9% | $23 | $0.40 |

| | Durability (per year) | Availability "designed" | Availability SLA | Storage (per TB per month) | GET retrieve milli |
|---|---|---|---|---|---|
| EBS | 99.8% | Unstated | 99.99% | $25/$45/**$100**/$125+ (sc1/st1/**gp2**/io1) | |
| EFS | "High" | "High" | – | $300 | |
| EC2 d2.xlarge instance store | Unstated | Unstated | – | $25.44 | $0 |

Especially notable items are in **boldface**. Sources: S3 pricing, S3 SLA, S3 FAQ, RRS info (note that this is considered d
EBS availability and durability, EBS pricing, EFS pricing, EC2 SLA

# EC2

## EC2 Basics

- 📒 Homepage · Documentation · FAQ · Pricing (see also ec2instances.info)
- **EC2** (Elastic Compute Cloud) is AWS' offering of the most fundamental piece of cloud computing: A virtual priva
can run most Linux, BSD, and Windows operating systems. Internally, they've used a heavily modified Xen virtua
instance classes are being introduced with a KVM derived hypervisor instead, called Nitro. So far, this is limited t
types. Lastly, there's a "bare metal hypervisor" available for i3.metal instances
- The term "EC2" is sometimes used to refer to the servers themselves, but technically refers more broadly to a wh
supporting services, too, like load balancing (CLBs/ALBs/NLBs), IP addresses (EIPs), bootable images (AMIs), secu
drives (EBS) (which we discuss individually in this guide).
- 💸 **EC2 pricing** and **cost management** is a complicated topic. It can range from free (on the AWS free tier) to a l
usage. Pricing is by instance type, by second or hour, and changes depending on AWS region and whether you
instances On-Demand, on the Spot market or pre-purchasing (Reserved Instances).
- **Network Performance:** For some instance types, AWS uses general terms like Low, Medium, and High to refer to
Users have done benchmarking to provide expectations for what these terms can mean.

Back to top ⬆️

## EC2 Alternatives and Lock-In

- Running EC2 is akin to running a set of physical servers, as long as you don't do automatic scaling or tooled clus set of static instances, migrating to another VPS or dedicated server provider should not be too hard.
- 🚪 **Alternatives to EC2:** The direct alternatives are Google Cloud, Microsoft Azure, Rackspace, DigitalOcean, AWS and other VPS providers, some of which offer similar APIs for setting up and removing instances. (See the comp
- **Should you use Amazon Linux?** AWS encourages use of their own Amazon Linux, which is evolved from Red Ha and CentOS. It's used by many, but others are skeptical. Whatever you do, think this decision through carefully. heavily tested and better supported in the unlikely event you have deeper issues with OS and virtualization on E companies do just fine using a standard, non-Amazon Linux distribution, such as Ubuntu or CentOS. Using a sta means you have an exactly replicable environment should you use another hosting provider instead of (or in ad helpful if you wish to test deployments on local developer machines running the same standard Linux distributio more common with Docker, too. Amazon now supports an official Amazon Linux Docker image, aimed at assisti on a comparable environment, though this is new enough that it should be considered experimental). Note that Amazon Linux 2 supports on-premise deployments explicitly.
- **EC2 costs:** See the section on this.

Back to top ⬆️

## EC2 Tips

- 🔹 **Picking regions:** When you first set up, consider which regions you want to use first. Many people in North A set up in the us-east-1 (N. Virginia) region, which is the default, but it's worth considering if this is best up front. service availability (some services are not available in all regions), costing (baseline costs also vary by region by u lowest in us-east-1 for comparison purposes)), and compliance (various countries have differing regulations with example).

- **Instance types:** EC2 instances come in many types, corresponding to the capabilities of the virtual machine in CP RAM, disk sizes and types (SSD or magnetic), and network bandwidth.

    - Selecting instance types is complex since there are so many types. Additionally there are different generatio
    - 🔹 Use the list at **ec2instances.info** to review costs and features. Amazon's own list of instance types is harc features and price together, which makes it doubly difficult.
    - Prices vary a lot, so use **ec2instances.info** to determine the set of machines that meet your needs and **ec2p** cheapest type in the region you're working in. Depending on the timing and region, it might be much chea *more* memory or CPU than the bare minimum.

- **Turn off** your instances when they aren't in use. For many situations such as testing or staging resources, you ma on 24/7, and you won't need to pay EC2 running costs when they are suspended. Given that costs are calculated simple mechanism for cost savings. This can be achieved using Lambda and CloudWatch, deploying the Instance open source option like cloudcycler, or a SaaS provider like GorillaStack. (Note: if you turn off instances with an any state will be lost when the instance is turned off. Therefore, for stateful applications it is safer to turn off EBS

- **Dedicated instances** and **dedicated hosts** are assigned hardware, instead of usual virtual instances. They are mo instances but can be preferable for performance, compliance, financial modeling, or licensing reasons.

- **32 bit vs 64 bit:** A few micro, small, and medium instances are still available to use as 32-bit architecture. You'll b ("amd64") instances nowadays, though smaller instances still support 32 bit ("i386"). Use 64 bit unless you have good reasons to use 32.

- **HVM vs PV:** There are two kinds of virtualization technology used by EC2, hardware virtual machine (HVM) and Historically, PV was the usual type, but now HVM is becoming the standard. If you want to use the newest instan HVM. See the instance type matrix for details.

- **Operating system:** To use EC2, you'll need to pick a base operating system. It can be Windows or Linux, such as You do this with AMIs, which are covered in more detail in their own section below.

- **Limits:** You can't create arbitrary numbers of instances. Default limits on numbers of EC2 instances per account v described in this list.

- ❗ **Use termination protection:** For any instances that are important and long-lived (in particular, aren't part of a termination protection. This is an important line of defense against user mistakes, such as accidentally terminatin of just one due to human error.

- **SSH key management:**
  - When you start an instance, you need to have at least one ssh key pair set up, to bootstrap, i.e., allow you to
  - Aside from bootstrapping, you should manage keys yourself on the instances, assigning individual keys to i as appropriate.
  - Avoid reusing the original boot keys except by administrators when creating new instances.
  - Avoid sharing keys and add individual ssh keys for individual users.

- **GPU support:** You can rent GPU-enabled instances on EC2 for use in machine learning or graphics rendering wo
  - There are three types of GPU-enabled instances currently available:
    - The P3 series offers NVIDIA Tesla V100 GPUs in 1, 4 and 8 GPU configurations targeting machine learni and other high performance computing applications.
    - The P2 series offers NVIDIA Tesla K80 GPUs in 1, 8 and 16 GPU configurations targeting machine learni and other high performance computing applications.
    - The G3 series offers NVIDIA Tesla M60 GPUs in 1, 2, or 4 GPU configurations targeting graphics and vid
  - AWS offers two different AMIs that are targeted to GPU applications. In particular, they target deep learning provide access to more stripped-down driver-only base images.
    - AWS offers both an Amazon Linux Deep Learning AMI (based on Amazon Linux) as well as an Ubuntu D come with most NVIDIA drivers and ancillary software (CUDA, CUBLAS, CuDNN, TensorFlow, PyTorch, e barrier to usage.
    - ⛓️ Note that using these AMIs can lead to lock in due to the fact that you have no direct access to soft versioning.
    - 🔸 The compendium of frameworks included can lead to long instance startup times and difficult-to-re
  - 🔹 As with any expensive EC2 instance types, Spot instances can offer significant savings with GPU workload tolerable.

- All current EC2 instance types can take advantage of IPv6 addressing, so long as they are launched in a subnet w range in an IPv6-enabled VPC.

Back to top ⬆️

## EC2 Gotchas and Limitations

- ❗ Never use ssh passwords. Just don't do it; they are too insecure, and consequences of compromise too sever
  up on this and fully disable ssh password access to your ssh server by making sure 'PasswordAuthentication no'
  /etc/ssh/sshd_config file. If you're careful about managing ssh private keys everywhere they are stored, it is a ma
  security over password-based authentication.
- 🔸 For all newer instance types, when selecting the AMI to use, be sure you select the HVM AMI, or it just won't
- ❗ When creating an instance and using a new ssh key pair, make sure the ssh key permissions are correct.
- 🔸 Sometimes certain EC2 instances can get scheduled for retirement by AWS due to "detected degradation of t
  in which case you are given a couple of weeks to migrate to a new instance
  - If your instance root device is an EBS volume, you can typically stop and then start the instance which move
    hardware, giving you control over timing of this event. Note however that you will lose any instance store v
    drives) if your instance type has instance store volumes.
  - The instance public IP (if it has one) will likely change unless you're using Elastic IPs. This could be a problem
    on the IP address.
- 🔸 Periodically you may find that your server or load balancer is receiving traffic for (presumably) a previous EC2
  the same IP address that you are handed out now (this may not matter, or it can be fixed by migrating to anothe
- ❗ If the EC2 API itself is a critical dependency of your infrastructure (e.g. for automated server replacement, cus
  etc.) and you are running at a large scale or making many EC2 API calls, make sure that you understand when th
  rate limited and the limits are not published and subject to change) and code and test against that possibility.
- ❗ Many newer EC2 instance types are either EBS-only, or backed by local NVMe disks assigned to the instance.
  performance and costs when planning to use them.
- ❗ If you're operating at significant scale, you may wish to break apart API calls that enumerate all of your resou
  either on individual resources, or a subset of the entire list. EC2 APIs will time out! Consider using filters to restri
- ❗ ⏱️ Instances come in two types: **Fixed Performance Instances** (e.g. M3, C3, and R3) and **Burstable Performa**
  T2 instance receives CPU credits continuously, the rate of which depends on the instance size. T2 instances accru
  are idle, and use CPU credits when they are active. However, once an instance runs out of credits, you'll notice a
  performance. If you need consistently high CPU performance for applications such as video encoding, high volu
  applications, it is recommended to use Fixed Performance Instances.

- Instance user-data is limited to 16 KB. (This limit applies to the data in raw form, not base64-encoded form.) If m
  be downloaded from S3 by a user-data script.
- Very new accounts may not be able to launch some instance types, such as GPU instances, because of an initially
  zero. This limit can be raised by making a support request. See AWS Service Limits for the method to make the s
  this limit of zero is not currently documented.
- Since multiple AWS instances all run on the same physical hardware, early cloud adopters encountered what bec
  Neighbor problem. This feeling of not getting what you are paying for led to user frustration, however "steal" m
  describe what's actually happening based on a detailed explanation of how the kernel determine steal time. Avo
  affect your application in the cloud may be best handled by properly designing your cloud architecture.
- AWS introduced Dedicated Tenancy in 2011. This allows customers to have all resources from a single server. So
  solve the noisy neighbor problem since only that customer uses the CPU. This approach comes with a significan
  needed any type of maintenance. If a customer had 20 instances running using shared tenancy and one underly
  maintenance, only the instance on that server would go offline. If that customer had 20 instances running using
  the underlying server needs maintenance, all 20 instances would go offline.
- 🔸 Only **i3.metal** type instances providing an ability to run Android x86 emulators on AWS at the moment.

## CloudWatch

### CloudWatch Basics

- 📒 Homepage · Documentation · FAQ · Pricing
- **CloudWatch** monitors resources and applications, captures logs, and sends events.
- CloudWatch monitoring is the standard mechanism for keeping tabs on AWS resources. A wide range of **metrics**
  available via CloudWatch, allowing you to create time based graphs, **alarms**, and **dashboards**.
  - Alarms are the most practical use of CloudWatch, allowing you to trigger notifications from any given metri
  - Alarms can trigger SNS notifications, Auto Scaling actions, or EC2 actions.
  - Alarms also support alerting when any M out of N datapoints cross the alarm threshold.
  - Publish and share graphs of metrics by creating customizable dashboard views.
    - Monitor and report on EC2 instance system check failure alarms.

- **Using CloudWatch Events:**
  - Events create a mechanism to automate actions in various services on AWS. You can create event rules from Auto Scaling, Run commands, deployments or time-based schedules (think Cron).
  - Triggered events can invoke Lambda functions, send SNS/SQS/Kinesis messages, or perform instance actions or snapshot volumes).
  - Custom payloads can be sent to targets in JSON format, this is especially useful when triggering Lambdas.
- **Using CloudWatch Logs:**
  - CloudWatch Logs is a streaming log storage system. By storing logs within AWS you have access to unlimited also have the option of streaming logs directly to ElasticSearch or custom Lambdas.
  - A log agent installed on your servers will process logs over time and send them to CloudWatch Logs.
  - You can export logged data to S3 or stream results to other AWS services.
  - CloudWatch Logs can be encrypted using keys managed through KMS.
- **Detailed monitoring:** Detailed monitoring for EC2 instances must be enabled to get granular metrics, and is bill

Back to top ⬆️

## CloudWatch Alternatives and Lock-In

- CloudWatch offers fairly basic functionality that doesn't create significant (additional) AWS lock-in. Most of the r service can be obtained through APIs that can be imported into other aggregation or visualization tools or servi provide CloudWatch data import services).
- 🚪 Alternatives to CloudWatch monitoring services include NewRelic, Datadog, Sumo Logic, Zabbix, Nagios, Ru source options such as StatsD or collectd with Graphite, and many others.
- 🚪 CloudWatch Log alternatives include Splunk, Sumo Logic, Loggly, LogDNA, Logstash, Papertrail, Elastic Stack logging solutions.

Back to top ⬆️

## CloudWatch Tips

- Some very common use cases for CloudWatch are **billing alarms**, **instance or load balancer up/down alarms**, a

- You can use EC2Config to monitor watch memory and disk metrics on Windows platform instances. For Linux, th
  that do the same thing.
- You can publish your own metrics using the AWS API. Incurs additional cost.
- You can stream directly from CloudWatch Logs to a Lambda or ElasticSearch cluster by creating subscriptions or
- Don't forget to take advantage of the CloudWatch non-expiring free tier.

Back to top ⬆️

### CloudWatch Gotchas and Limitations

- 🔸 Metrics in CloudWatch originate on the hypervisor. The hypervisor doesn't have access to OS information, so
  notably memory utilization) are not available unless pushed to CloudWatch from inside the instance.
- 🔸 You can not use more than one metric for an alarm.
- 🔸 Notifications you receive from alarms will not have any contextual detail; they have only the specifics of the t
  timing.
- 🔸 By default, CloudWatch metric resolution is 1 minute. If you send multiple values of a metric within the same
  aggregated into minimum, maximum, average and total (sum) per minute.
- 🐥 In July 2017, a new high-resolution option was added for CloudWatch metrics and alarms. This feature allows
  1-second resolution, and to evaluate CloudWatch alarms every 10 seconds.
  - The blog post introducing this feature describes how to publish a high-resolution metric to CloudWatch. No
    `PutMetricData` API, `StorageResolution` is an attribute of each item you send in the `MetricData` array, not
    `PutMetricData` API call.
- 🔸 Data about metrics is kept in CloudWatch for 15 months, starting November 2016 (used to be 14 days). Minir
  after 15 days.

## AMIs

### AMI Basics

- 📒 User guide

- **AMIs** (Amazon Machine Images) are immutable images that are used to launch preconfigured EC2 instances. Th
  private flavors. Access to public AMIs is either freely available (shared/community AMIs) or bought and sold in t
- Many operating system vendors publish ready-to-use base AMIs. For Ubuntu, see the Ubuntu AMI Finder. Amaz
  Amazon Linux.

Back to top ⬆️

## AMI Tips

- AMIs are built independently based on how they will be deploy. You must select AMIs that match your deploy
  creating them:
  - EBS or instance store
  - PV or HVM virtualization types
  - 32 bit ("i386") vs 64 bit ("amd64") architecture
- As discussed above, modern deployments will usually be with **64-bit EBS-backed HVM**.
- You can create your own custom AMI by snapshotting the state of an EC2 instance that you have modified.
- AMIs backed by EBS storage have the necessary image data loaded into the EBS volume itself and don't require
  which results in EBS-backed instances coming up much faster than instance storage-backed ones.
- **AMIs are per region**, so you must look up AMIs in your region, or copy your AMIs between regions with the AM
- As with other AWS resources, it's wise to use tags to version AMIs and manage their lifecycle.
- If you create your own AMIs, there is always some tension in choosing how much installation and configuration
  them.
  - Baking less into your AMIs (for example, just a configuration management client that downloads, installs, ar
    new EC2 instances when they are launched) allows you to minimize time spent automating AMI creation an
    lifecycle (you will likely be able to use fewer AMIs and will probably not need to update them as frequently)
    before new instances are ready for use and results in a higher chance of launch-time installation or configur
  - Baking more into your AMIs (for example, pre-installing but not fully configuring common software along w
    management client that loads configuration settings at launch time) results in a faster launch time and fewe
    software installation and configuration to break at instance launch time but increases the need for you to cr
    AMI creation pipeline.

    - Baking even more into your AMIs (for example, installing all required software as well and potentially also e configuration information) results in fast launch times and a much lower chance of instance launch-time fai additional re-deployment and re-configuration considerations) can require time consuming AMI updates in or configuration as well as more complex AMI creation automation processes.
- Which option you favor depends on how quickly you need to scale up capacity, and size and maturity of your te
    - When instances boot fast, auto-scaled services require less spare capacity built in and can more quickly scal sudden increases in load. When setting up a service with autoscaling, consider baking more into your AMIs EBS storage option.
    - As systems become larger, it common to have more complex AMI management, such as a multi-stage AMI few (ideally one) common base AMIs are infrequently regenerated when components that are common to a updated and then a more frequently run "service-level" AMI generation process that includes installation an application-specific software.
- More thinking on AMI creation strategies here.
- Use tools like Packer to simplify and automate AMI creation.
- If you use RHEL instances and happen to have existing RHEL on-premise Red Hat subscriptions, then you could Access program to migrate a portion of your subscriptions to AWS, and thereby not having AWS charge you for second time. You can either use your own self-created RHEL AMI's or Red Hat provided Gold Images that will be AMI's once you sign up for Red Hat Cloud Access.

Back to top ⬆️

## AMI Gotchas and Limitations

- 🔸 **Amazon Linux package versions:** By default, instances based on Amazon Linux AMIs are configured point to packages in Amazon's package repository. This means that the package versions that get installed are not locke changes, including breaking ones, to appear when applying updates in the future. If you bake your AMIs with up is unlikely to cause problems in running services whose instances are based on those AMIs – breaks will appear stage of your build process, and will need to be fixed or worked around before new AMIs can be generated. The feature that allows you to configure Amazon Linux instances to target the repository of a particular major versio AMI, reducing the likelihood that breaks caused by Amazon-initiated package version changes will occur at pack cost of not having updated packages get automatically installed by future update runs. Pairing use of the "lock c process to advance the Amazon Linux AMI at your discretion can give you tighter control over update behaviors

- **Cloud-Init Defaults:** Oftentimes users create AMIs after performing customizations (albeit manually or via some Ansible). If you're not careful to alter cloud-init settings that correspond to the system service (e.g. sshd, etc.) yo find that your changes are no longer in effect after booting your new AMI for the first time, as cloud-init has ove

  Some distros have different files than others, but all are generally located in `/etc/cloud/`, regardless of distro. Y these files carefully for your chosen distro before rolling your own AMIs. A complete reference to cloud-init is av site. This is an advanced configuration mechanism, so test any changes made to these files in a sandbox prior to

## Auto Scaling

### Auto Scaling Basics

- 📒 Homepage · User guide · FAQ · Pricing at no additional charge
- Auto Scaling Groups (ASGs) are used to control the number of instances in a service, reducing manual effort to EC2 instances.
- They can be configured through Scaling Policies to automatically increase or decrease instance counts based on utilization, or based on a schedule.
- There are three common ways of using ASGs - dynamic (automatically adjust instance count based on metrics fo utilization), static (maintain a specific instance count at all times), scheduled (maintain different instance counts a on days of the week).
- 💸 ASGs have no additional charge themselves; you pay for underlying EC2 and CloudWatch services.

Back to top ⬆️

### Auto Scaling Tips

- 💸 Better matching your cluster size to your current resource requirements through use of ASGs can result in sig
many types of workloads.
- Pairing ASGs with CLBs is a common pattern used to deal with changes in the amount of traffic a service receive
- Dynamic Auto Scaling is easiest to use with stateless, horizontally scalable services.
- Even if you are not using ASGs to dynamically increase or decrease instance counts, you should seriously consid
inside of ASGs – given a target instance count, the ASG will work to ensure that number of instances running is e
replacing instances for you if they die or are marked as being unhealthy. This results in consistent capacity and b
service.
- Autoscalers can be configured to terminate instances that a CLB or ALB has marked as being unhealthy.

Back to top ⬆️

### Auto Scaling Gotchas and Limitations

- 🔸 **ReplaceUnhealthy setting:** By default, ASGs will kill instances that the EC2 instance manager considers to be
for instances whose CPU is completely saturated for minutes at a time to appear to be unresponsive, causing an
ReplaceUnhealthy setting turned on to replace them. When instances that are managed by ASGs are expected to
high CPU, consider deactivating this setting. If you do so, however, detecting and killing unhealthy nodes will be

## EBS

### EBS Basics

- 📒 Homepage · User guide · FAQ · Pricing
- **EBS** (Elastic Block Store) provides block level storage. That is, it offers storage volumes that can be attached as fi
network drives.

- EBS volumes can only be attached to one EC2 instance at a time. In contrast, EFS can be shared but has a much comparison).

Back to top ⬆️

## EBS Tips

- ⏱ **RAID:** Use RAID drives for increased performance.
- ⏱ A worthy read is AWS' post on EBS IO characteristics as well as their performance tips.
- ⏱ One can provision IOPS (that is, pay for a specific level of I/O operations per second) to ensure a particular le disk.
- ⏱ A single gp2 EBS volume allows 16k IOPS max To get the maximum performance out of a gp2 EBS volume, it size and attached to an EBS-optimized EC2 instance.
- 💸 Standard and gp2 EBS volumes improve IOPS with size. It may make sense for you to simply enlarge a volum better performance explicitly. This can in many cases reduce costs by 2/3.
- A standard block size for an EBS volume is 16kb.

Back to top ⬆️

## EBS Gotchas and Limitations

- ❗ EBS durability is reasonably good for a regular hardware drive (annual failure rate of between 0.1% - 0.2%). C very poor if you don't have backups! By contrast, S3 durability is extremely high. *If you care about your data, bac snapshots.*
- 🔸 EBS has an SLA with **99.99%** uptime. See notes on high availability below.
- ❗ EBS volumes have a **volume type** indicating the physical storage type. The types called "standard" (**st1** or **sc1** platter disks, which deliver only hundreds of IOPS — not what you want unless you're really trying to cut costs. M **io1** are typically the options you want.
- ❗ When restoring a snapshot to create an EBS volume, blocks are lazily read from S3 the first time they're refere period of high latency, you may wish to use `dd` or `fio` as per the official documentation.

# EFS

## EFS Basics

- 📒 Homepage · User guide · FAQ · Pricing
- 🐥 **EFS** is Amazon's network filesystem. It's presented as an NFSv4.1 server. Any compatible NFSv4 client can mo
- It is designed to be highly available and durable and each EFS file system object is redundantly stored across mu
- EFS is designed to be used as a shared network drive and it can automatically scale up to petabytes of stored da instances attached to it.
- EFS can offer higher throughput (multiple gigabytes per second) and better durability and availability than EBS ( but with higher latency.
- EFS is priced based on the volume of data stored, and costs much more than EBS; it's in the ballpark of three tim general purpose gp2 EBS volumes.
- ⏱️ Performance is dependent on the volume of data stored, as is the price:
  - Like EBS, EFS uses a credit based system. Credits are earned at a rate of 50 KiB/s per GiB of storage and con reading/writing files or metadata. Unlike EBS, operations on metadata (file size, owner, date, etc.) also consu BurstCreditBalance metric in CloudWatch should be monitored to make sure the file system doesn't run out
  - Throughput capacity during bursts is also dependent on size. Under 1 TiB, throughput can go up to 100 MiB is added for each stored TiB. For instance, a file system storing 5 TiB would be able to burst at a rate of 500 throughput per EC2 instance is 250 MiB/s.
  - EFS has two performance modes that can only be set when a file system is created. One is "General Purpose Max I/O scales higher, but at the cost of higher latency. When in doubt, use General Purpose, which is also t PercentIOLimit metric in CloudWatch hovers around 100%, Max I/O is recommended. Changing performanc new EFS and migrating data.
- High availability is achieved by having mount targets in different subnets / availability zones.

Back to top ⬆️

## EFS Tips

- With EFS being based on NFSv4.1, any directory on the EFS can be mounted directly, it doesn't have to be the ro application could mount *fs-12345678:/prog1*, another *fs-12345678:/prog2*.
- User and group level permissions can be used to control access to certain directories on the EFS file system.
- ⏱️ **Sharing EFS filesystems:** One EFS filesystem can be used for multiple applications or services, but it should b

  Pros:

  - Because performance is based on total size of stored files, having everything on one drive will increase perf application consuming credits faster than it can accumulate might be offset by another application that just rarely accesses them.

  Cons:

  - Since credits are shared, if one application over-consumes them, it will affect the others.
  - A compromise is made with regards to security: all clients will have to have network access to the drive. Son one client instance can mount any directory on the EFS and they have read-write access to all files on the dr access to the applications hosted on other clients. There isn't a no-root-squash equivalent for EFS.

Back to top ⬆️

## EFS Gotchas and Limitations

- 🔸 A number of NFSv4.1 features are not supported and there are some limits to the service.
- 🔸 As of 2017-08, EFS offers disk level encryption for new drives. For file systems created before that date, encry achieved by moving the data to a new EFS volume.
- 🔸 An EFS file system can be mounted on premises over Direct Connect.
- 🔸 An EFS file system can NOT be mounted over VPC peering or VPN, even if the VPN is running on top of Dire
- 🔸 Using an EFS volume on Windows is not supported.
- ⏱️ When a file is uploaded to EFS, it can take hours for EFS to update the details for billing and burst credit pur
- 🔸 ⏱️ Metadata operations can be costly in terms of burst credit consumption. Recursively traversing a tree con can easily ramp up to tens or even hundreds of megabytes of burst credits being consumed, even if no file is be

like `find` or `chown -R` can have an adverse impact on performance.

# Load Balancers

## Load Balancer Basics

- AWS has 3 load balancing products - "Classic Load Balancers" (CLBs), "Application Load Balancers" (ALBs), and "N
(NLB).
- Before the introduction of ALBs, "Classic Load Balancers" were known as "Elastic Load Balancers" (ELBs), so older
and blog posts may still reference "ELBs".
- CLBs have been around since 2009, ALBs in 2016, NLBs were added in 2017 to AWS.
- CLBs support TCP and HTTP load balancing. ALBs support HTTP load balancing only. NLBs support TCP layer 4 lo
- CLBs and ALBs can optionally handle termination for a single SSL certificate.
- All can optionally perform active health checks of instances and remove them from the destination pool if they b
- CLBs don't support complex / rule-based routing. ALBs support a (currently small) set of rule-based routing feat
extensive routing options.
- CLBs can only forward traffic to a single globally configured port on destination instances, while ALBs can forwar
configured on a per-instance basis, better supporting routing to services on shared clusters with dynamic port a
Mesos). NLBs support multiple ports on same IP; registering targets by IP address, including targets outside the
ECS can select unused port for scheduling a task then register a target group using this port.
- CLBs are supported in EC2 Classic as well as in VPCs while ALBs are supported in VPCs only.
- ALBs can target groups of instances and IP based targets in the RFC1918 ranges allowing you to use on premise
Direct Connect.

Back to top ⬆️

## Load Balancer Tips

- If you don't have opinions on your load balancing up front, and don't have complex load balancing needs like a
of requests, it's reasonable just to use a CLB or ALB for load balancing instead.

- Even if you don't want to think about load balancing at all, because your architecture is so simple (say, just one s in front of it anyway. This gives you more flexibility when upgrading, since you won't have to change any DNS se propagate, and also it lets you do a few things like terminate SSL more easily.
- **CLBs and ALBs have many IPs:** Internally, an AWS load balancer is simply a collection of individual software load EC2, with DNS load balancing traffic among them. The pool can contain many IPs, at least one per availability zo traffic levels. They also support SSL termination, which is very convenient.
- **Scaling:** CLBs and ALBs can scale to very high throughput, but scaling up is not instantaneous. If you're expectin traffic suddenly, it can make sense to load test them so they scale up in advance. You can also contact Amazon a the load balancer.
- **Client IPs:** In general, if servers want to know true client IP addresses, load balancers must forward this informat the standard X-Forwarded-For header. When using a CLB as an HTTP load balancer, it's possible to get the clien
- **Using load balancers when deploying:** One common pattern is to swap instances in the load balancer after spin your latest version, keep old stack running for one or two hours, and either flip back to old stack in case of prob
- **Rotating Certificates while retaining ARN:** Rotating IAM Server Certificates can be difficult as the standard prac then update all resources with the new ARN. You can however retain the same ARN using the `update-certifica` process:

1. Upload a new IAM Server Certificate with a unique name (e.g fuzzy.com.new)
2. Rename the existing IAM Server Certificate (e.g fuzzy.com to fuzzy.com.expired)
3. Rename the new IAM Server Certificate to the name of the previously existing certificate (e.g fuzzy.com.new to f
4. Jiggle the CLB/ALB Listener to pick up the change:
    - ALB: Invoke modify-listener with the existing details for the ALB Listener
    - CLB: Invoke create-load-balancer-listeners with the existing details for the CLB listener

Back to top ⬆️

## Load Balancer Gotchas and Limitations

- ❗ CLBs and ALBs have **no fixed external IP** that all clients see. For most consumer apps this doesn't matter, but yours may want this. IPs will be different for each user, and will vary unpredictably for a single client over time (w ranges). And similarly, never resolve a CLB name to an IP and put it as the value of an A record — it will work for

- ❗ Some web clients or reverse proxies cache DNS lookups for a long time, which is problematic for CLBs and A IPs. This means after a few minutes, hours, or days, your client will stop working, unless you disable DNS caching settings and be sure to adjust them properly. Another example is nginx as a reverse proxy, which normally resol up (although there is a way to get around this).
- ❗ It's not unheard of for IPs to be recycled between customers without a long cool-off period. So as a client, if not using SSL (to verify the server), you might get not just errors, but responses from completely different servic
- 🔸 As an operator of a service behind a CLB or ALB, the latter phenomenon means you can also see puzzling or clients of other companies. This is most common with clients using back-end APIs (since web browsers typically
- ❗ CLBs and ALBs take time to scale up, it does not handle sudden spikes in traffic well. Therefore, if you anticipa "pre-warm" the load balancer by gradually sending an increasing amount of traffic.
- ❗ Tune your healthchecks carefully — if you are too aggressive about deciding when to remove an instance an adding it back into the pool, the service that your load balancer is fronting may become inaccessible for second extra careful about this when an autoscaler is configured to terminate instances that are marked as being unhea balancer.
- ❗ CLB HTTPS listeners don't support Server Name Indication (SNI). If you need SNI, you can work around this li providing a certificate with Subject Alternative Names (SANs) or by using TCP listeners and terminating SSL at yo
- 🔸 There is a limit on the number of ALBs, CLBs and NLBs per region (separately). As of late 2017, the default lim region. These limits can be easily raised for ALB and CLB, but AWS is quite reluctant to raise the limit on NLBs.
- 🔸 If using a Network Load Balancer (NLB) then EC2 clients cannot connect to an NLB that resides in another VP managed VPN unless the EC2 client is a C5, i3.metal or M5 instance type. For VPC peering, both VPCs must be in Troubleshooting.

## CLB

### CLB Basics

- 📒 Homepage · User guide · FAQ · Pricing
- Classic Load Balancers, formerly known as Elastic Load Balancers, are HTTP and TCP load balancers that are man Amazon.

Back to top ⬆️

## CLB Tips

- **Best practices:** This article is a must-read if you use CLBs heavily, and has a lot more detail.

Back to top ⬆️

## CLB Gotchas and Limitations

- In general, CLBs are not as "smart" as some load balancers, and don't have fancy features or fine-grained contro load balancer would offer. For most common cases involving sessionless apps or cookie-based sessions over HT they work well.
- 🔸 By default, CLBs will refuse to route traffic from a load balancer in one Availability Zone (AZ) to a backend ins cause 503s if the last instance in an AZ becomes unavailable, even if there are healthy instances in other zones. I two backend instances per AZ, you almost certainly want to enable cross-zone load balancing.
- 🔸 Complex rules for directing traffic are not supported. For example, you can't direct traffic based on a regular HAProxy offers.
- **Apex DNS names:** Once upon a time, you couldn't assign a CLB to an apex DNS record (i.e. example.com instead because it needed to be an A record instead of a CNAME. This is now possible with a Route 53 alias record direc balancer.
- 🔸 CLBs use HTTP keep-alives on the internal side. This can cause an unexpected side effect: Requests from diffe own TCP connection on the external side, can end up on the same TCP connection on the internal side. Never as requests on the same TCP connection are from the same client!
- 🔸 Traffic between CLBs and back-end instances in the same subnet **will** have Network ACL rules evaluated (EC2 subnet would not have Network ACL rules evaluated). If the default '0.0.0.0/0 ALLOW' rule is removed from the N subnet, a rule that allows traffic on both the health check port and any listener port must be added.
- As of December 2016, CLBs launched in VPCs do not support IPv6 addressing. CLBs launched in EC2-Classic sup with the "dualstack" DNS name.

# ALB

## ALB Basics

- 📒 Homepage · User guide · FAQ · Pricing
- 🐥 **Websockets and HTTP/2** are now supported.
- 🐥 **Internet Protocol Version 6 (IPv6)** is now supported.
- 🐥 **Load Balancing via IP** is now supported.
- Prior to the Application Load Balancer, you were advised to use TCP instead of HTTP as the protocol to make it v
  and use the obscure but useful Proxy Protocol (more on this) to pass client IPs over a TCP load balancer.

Back to top ⬆️

## ALB Tips

- Use ALBs to route to services that are hosted on shared clusters with dynamic port assignment (like ECS or Mesc
- ALBs support HTTP host-based routing (send HTTP requests for "api.mydomain.com" -> {target-group-1}, "blog
  group 2}) as well as HTTP path-based routing (send HTTP requests for "/api/*" -> {target-group-1}, "/blog/*" ->

Back to top ⬆️

## ALB Gotchas and Limitations

- 🔸 ALBs only support HTTP/2 over HTTPS (no plain-text HTTP/2).
- 🔸 ALBs only support HTTP/2 to external clients and not to internal resources (instances/containers).
- ALBs support HTTP routing but not port-based TCP routing.
- Instances in the ALB's target groups have to either have a single, fixed healthcheck port ("EC2 instance"-level he
  healthcheck port for a target has to be the same as its application port ("Application instance"-level healthcheck
  per-target healthcheck port that is different than the application port.
- ALBs are VPC-only (they are not available in EC2 Classic)

- In a target group, if there is no healthy target, all requests are routed to all targets. For example, if you point a lis containing a single service that has a long initialization phase (during which the health checks would fail), reques while it is still starting up.
- 📜 Although ALBs now support SNI, they only support 25 HTTPS certificates per Load Balancer. This limitation is might be subject to change.

## Elastic Beanstalk

### Elastic Beanstalk Basics

- 📒 Homepage · Developer guide · FAQ · Pricing
- **EB** (Elastic Beanstalk) is a PaaS (Platform as a Service) that helps developers create, deploy and scale web applica
- EB handles deployment, configuration, provisioning, load balancing, auto-scaling, monitoring, and logging
- EB creates AWS resources on your behalf but you retain full access and control of the underlying resources
- 💸 There is no cost to use EB but you will still be charged the full cost of the underlying AWS resources created

Back to top ⬆️

### Elastic Beanstalk Tips

- To speed up deployment before launch or in a dev stage, turn off health checks and set the `Deployment policy`
- If you have a configuration you want to re-use for multiple EB apps, you can save the current configuration using `myEBConfig`
- By default, EB doesn't have any alarms. You'll need to add them yourself on metrics that you're monitoring.
- By default, EB doesn't enable managed platform updates. Enable them in configuration to have EB automatically pre-specified maintenance window

Back to top ⬆️

### Elastic Beanstalk Gotchas and Limitations

- 🔸 Don't edit [apache|nginx] conf files manually on ec2 instances as they will be re-written on each deployment
- 🔸 After creating an EB environment, it's no longer possible to change the `Name` tag
- 🔸 EB will sometimes quarantine instances that cause multiple deployment issues. Despite being quarantined, E on subsequent deployments. To prevent this behavior, said instances will need to be terminated (or the underlyi
- File uploads are capped at 10MB for most default eb configurations - update nginx config to change
- If you edit `.elasticbeanstalk/saved_configs/`, be aware that this is not kept in sync with the EB environment co manually fetch and save for changes to take effect

## Elastic IPs

### Elastic IP Basics

- 📒 Documentation · FAQ · Pricing
- **Elastic IPs** are static IP addresses you can rent from AWS to assign to EC2 instances.

Back to top ⬆️

### Elastic IP Tips

- 🔹 **Prefer load balancers to elastic IPs:** For single-instance deployments, you could just assign elastic IP to an in name, and consider that your deployment. Most of the time, you should provision a load balancer instead:
  - It's easy to add and remove instances from load balancers. It's also quicker to add or remove instances from reassign an elastic IP.
  - It's more convenient to point DNS records to load balancers, instead of pointing them to specific IPs you m also be Route 53 aliases, which are easier to change and manage.
  - But in some situations, you do need to manage and fix IP addresses of EC2 instances, for example if a custo situations require elastic IPs.
- Elastic IPs are limited to 5 per account. It's possible to request more.
- If an Elastic IP is not attached to an active resource there is a small hourly fee.

- Elastic IPs are no extra charge as long as you're using them. They have a (small) cost when not in use, which is a people from squatting on excessive numbers of IP addresses.

Back to top ⬆️

### Elastic IP Gotchas and Limitations

- 🔸 There is officially no way to allocate a contiguous block of IP addresses, something you may desire when givi Though when allocating at once, you may get lucky and have some be part of the same CIDR block. If this is imp want to bring your own IP, which is more involved than this guide will go into.
- Unofficially, if you have Enterprise support, you can ask your account rep to try to allocate a block of Elastic IPs v For example, some of Duo's fixed ranges are blocks of AWS IP space reassigned to `AWS-DUOSECURITYINC` . This is expect a denial.

## Glacier

### Glacier Basics

- 📒 Homepage · Developer guide · FAQ · Pricing
- **Glacier** is a lower-cost alternative to S3 when data is infrequently accessed, such as for archival purposes.
- It's only useful for data that is rarely accessed. It generally takes 3-5 hours to fulfill a retrieval request.
- AWS has not officially revealed the storage media used by Glacier; it may be low-spin hard drives or even tapes.
- AWS has released an even more cost effective storate tier called Glacier Deep Archive that offers ~12 hour retrie roughly a thousand dollars per month per petabyte.

Back to top ⬆️

### Glacier Tips

- You can physically ship your data to Amazon to put on Glacier on a USB or eSATA HDD.

Back to top ⬆️

### Glacier Gotchas and Limitations

- 🔸 Getting files off Glacier is glacially slow (typically 3-5 hours or more).
- 🔸 Due to a fixed overhead per file (you pay per PUT or GET operation), uploading and downloading many smal
  be very expensive. There is also a 32k storage overhead per file. Hence it's a good idea is to archive files before
- 💸 Be aware of the per-object costs of archiving S3 data to Glacier. It costs $0.05 per 1,000 requests. If you have
  objects of relatively small size, it will take time to reach a break-even point (initial archiving cost versus lower st

## Quicksight

### Quicksight Basics

- 📒 Homepage · User guide · Pricing

Back to top ⬆️

### Quicksight Gotchas and Limitations

- ❗ Out of the box Quicksight is not able to access tables that are linked to a Schema in the AWS Glue Schema R
  auto-generated IAM role `aws-quicksight-service-role-v0` does not have the necessary permissions. You can't
  used but you can add more permissions to the role. The error message you will receive is an `SQL_EXCEPTION` wit
  `line 2:8: Column 'columnname' cannot be resolved` where `columnname` is the first column in your table.
- 🔸 Only QuickSight accounts that were created in the US East (N. Virginia) region can access the QuickSight Foru
  can only have a QuickSight account in one region per AWS account.

## RDS

### RDS Basics

- 📒 Homepage · User guide · FAQ · Pricing (see also ec2instances.info/rds/)

- **RDS** is a managed relational database service, allowing you to deploy and scale databases more easily. It suppor Server, PostgreSQL, MySQL, MariaDB, and Amazon's own Aurora.
- RDS offers out of the box support for high availability and failover for your databases.

Back to top ⬆️

## RDS Tips

- If you're looking for the managed convenience of RDS for other data stores such as MongoDB or Cassandra, you third-party services from providers such as [Compose, or InstaClustr.
- 🔹 Make sure to create a new parameter group and option group for your database since the default parameter dynamic configuration changes.
- RDS instances start with a default timezone of UTC. If necessary, this can be changed to a different timezone.
- Save costs by stopping the RDS instance(s) when not in use, you can achieve this by configuring a stop and start Instance Scheduler solution.

Back to top ⬆️

## RDS Gotchas and Limitations

- ⏱️ RDS instances run on EBS volumes (either general-purpose or provisioned IOPS), and hence are constrained
- 🔸 Verify what database features you need, as not everything you might want is available on RDS. For example, i check the list of supported features and extensions. If the features you need aren't supported by RDS, you'll hav yourself.
- 🔸 If you use the failover support offered by RDS, keep in mind that it is based on DNS changes, and make sure these changes appropriately. This is particularly important for Java, given how its DNS resolver's TTL is configure
- 🔸 **DB migration to RDS:** While importing your database into RDS ensure you take into consideration the maint a backup is running at the same time, your import can take a considerably longer time than you would have exp
- Database sizes are limited to **6TB** for all database engines except for SQL Server which has a **4TB** limit and Auro **64TB** databases.

# RDS MySQL and MariaDB

## RDS MySQL and MariaDB Basics

- RDS offers MySQL versions 5.5, 5.6, 5.7 and 5.8.

- RDS offers MariaDB versions 10.0, 10.1, 10.2 and 10.3.

Back to top ⬆️

## RDS MySQL and MariaDB Tips

- MySQL RDS allows access to binary logs.
- Multi-AZ instances of MySQL transparently replicate data across AZs using DRBD. Automated backups of multi-
  backup instance to reduce latency spikes on the primary.
- 🔸 **Performance Schema:** While Performance Schema is enabled by default in MySQL 5.6.6 and later, it is disable
  of RDS. If you wish to enable Performance Schema, a reboot of the RDS instance will be required.
- 🔸 **MySQL vs MariaDB vs Aurora:** If you prefer a MySQL-style database but are starting something new, you pro
  Aurora and MariaDB as well. **Aurora** has increased availability and is the next-generation solution. That said, Aur
  faster than MySQL for certain workloads. **MariaDB**, the modern community fork of MySQL, likely now has the ec
  purposes and is supported by RDS.

Back to top ⬆️

## RDS MySQL and MariaDB Gotchas and Limitations

- 🔸 **No SUPER privileges.** RDS provides some stored procedures to perform some tasks that require SUPER privil
  stopping replication.
- 🔸 You can replicate to non-RDS instances of MySQL, but replication to these instances will break during AZ fail

- 🔸 There is no ability to manually CHANGE MASTER on replicas, so they must all be rebuilt after a failover of the
- 🔸 Most global options are exposed only via DB parameter groups. Some variables that were introduced in later as avoid_temporal_upgrade in MySQL 5.6.24 are not made available in RDS's 5.6.x parameter group and making upgrade to MySQL 5.7.x.
- 🔸 RDS features such as Point-In-Time restore and snapshot restore are not supported on MyISAM tables. Ensur MyISAM table before executing a snapshot or backup operation to ensure consistency.

## RDS PostgreSQL

### RDS PostgreSQL Basics

- RDS offers PostgreSQL 9.3, 9.4, 9.5, 9.6, and 10.

Back to top ⬆️

### RDS PostgreSQL Tips

- Recently Logical Replication is being supported, both as subscriber and publisher.
- Supports a relatively large range of native extensions.
- RDS PostgreSQL 10 Supports native partitioning and most of the major features and tunables.
- Supports connections over SSL.
- Supports multi A-Z and Point-in-time recovery.

Back to top ⬆️

### RDS PostgreSQL Gotchas and Limitations

- No superuser privileges. RDS provides a role `rds_superuser` that can do most of the needed operations but the
- Some major features are delayed compared to open source PostgreSQL.
- By default RDS is spec'd with general purpose SSD , if you need better performance you have to spec provisione
- You can't use RDS as a replica outside RDS without using logical replication.

- There are settings that cannot be changed and most of the settings that can change can only be changed using groups.
- It's harder to troubleshoot performance problems since you have no access to the host.
- Be sure to verify that all the extensions you need are available. If you are using an extension not listed there, you a work around, or deploy your own database in EC2.
- Many Postgres utilities and maintenance items expect command line access, that can usually be satisfied by usin

## RDS SQL Server

### RDS SQL Server Basics

- RDS offers SQL Server 2008 R2, 2012, 2014, 2016 and 2017 including Express, Web, Standard and Enterprise.

Back to top ⬆️

### RDS SQL Server Tips

- Recently added support for backup and restore to/from S3 which may make it an attractive DR option for on-pr

Back to top ⬆️

### RDS SQL Server Gotchas and Limitations

- 🔸 The user is granted only db_owner privileges for each database on the instance.
- 🔸 Storage cannot be expanded for existing databases. If you need more space, you must restore your database larger storage.
- 🔸 There is a **16TB** database size limit for non-Express editions. There is also a minimum storage size, 20GB for W Standard and Enterprise.
- 🔸 Limited to 30 databases per instance

# RDS Aurora

## RDS Aurora Basics

Aurora is a cloud only database service designed to provide a distributed, fault-tolerant relational database with self-
scaling up to 64TB per instance. It currently comes in two versions, a MySQL compatible system, and a PostgreSQL c

# RDS Aurora MySQL

## RDS Aurora MySQL Basics

- Amazon's proprietary fork of MySQL intended to scale up for high concurrency workloads. Generally speaking, i
  performance under Aurora is not expected to improve significantly relative to MySQL or MariaDB, but Aurora is
  performance while executing many more queries concurrently than an equivalent MySQL or MariaDB server cou
- Notable new features include:
  - Log-structured storage instead of B-trees to improve write performance.
  - Out-of-process buffer pool so that databases instances can be restarted without clearing the buffer pool.
  - The underlying physical storage is a specialized SSD array that automatically maintains 6 copies of your data
  - Aurora read replicas share the storage layer with the write master which significantly reduces replica lag, eli
    master to write and distribute the binary log for replication, and allows for zero-data-loss failovers from the
    master and all the read replicas that share storage are known collectively as an **Aurora cluster**. Read replica

Back to top ⬆️

## RDS Aurora MySQL Tips

- In order to take advantage of Aurora's higher concurrency, applications should be configured with large databas
  should execute as many queries concurrently as possible. For example, Aurora servers have been tested to prod
  on some OLTP workloads with up to 5,000 connections.
- Aurora scales well with multiple CPUs and may require a large instance class for optimal performance.

- The easiest migration path to Aurora is restoring a database snapshot from MySQL 5.6 or 5.7. The next easiest m from a MySQL-compatible database such as MariaDB. For low-downtime migrations from other MySQL-compat up an Aurora instance as a replica of your existing database. If none of those methods are options, Amazon offe migration service.
- You can replicate from an Aurora cluster to MySQL or to another Aurora cluster. This requires binary logging to performant as native Aurora replication.
- Because Aurora read replicas are the equivalent of a multi-AZ backup and they can be configured as zero-data- are fewer scenarios in which the creation of a multi-AZ Aurora instance is required.

Back to top ⬆️

### RDS Aurora MySQL Gotchas and Limitations

- 🔸 Aurora 1.x is based on MySQL 5.6.x with some cherry-picking of later MySQL features. It is missing most 5.7 f online DDL features introduced in 5.6.17.
- 🔸 Aurora 2.x is based on MySQL 5.7.x
- Aurora does not support GTID transactions in either the 5.6/Aurora 1.x or the 5.7/Aurora 2.x release lines.
- Aurora maximum cluster size is 64 TB

## RDS Aurora PostgreSQL

### RDS Aurora PostgreSQL Basics

- Amazon's proprietary fork of PostgreSQL, intended to scale up for high concurrency workloads while maintainin based on PostgreSQL 9.6.
- Higher throughput (up to 3x with similar hardware).
- Automatic storage scale in 10GB increments up to 64TB.
- Low latency read replicas that share the storage layer with the master which significantly reduces replica lag.
- Point in time recovery.
- Fast database snapshots.

Back to top ⬆️

### RDS Aurora PostgreSQL Tips

- Aurora Postgres by default is supposed to utilize high connection rates and for this reason connection pooling n accordingly.
- Because Aurora is based on PostgreSQL 9.6, it lacks features like declarative partitioning or logical replication.

Back to top ⬆️

### RDS Aurora PostgreSQL Gotchas and Limitations

- Aurora PostgreSQL falls behind normal RDS when it comes to available versions, so if you need features from th you might be better off with plain RDS.
- Patching and bug fixing is separate from open source PostgreSQL.

## ElastiCache

### ElastiCache Basics

- 📒 Homepage · User guide for Redis · User guide for Memcached · FAQ · Pricing
- **ElastiCache** is a managed in-memory cache service, that can be used to store temporary data in a fast in-memo to avoid repeating the same computation multiple times when it could be reused.
- It supports both the Memcached and Redis open source in-memory cache software and exposes them both usin
- The main benefit is that AWS takes care of running, patching and optimizing the cache nodes for you, so you jus and configure its endpoint in your application, while AWS will take of most of the operational work of running th

Back to top ⬆️

### ElastiCache Tips

- Choose the engine, clustering configuration and instance type carefully based on your application needs. The do
  detail the pros, cons and limitations of each engine in order to help you choose the best fit for your application.
  preferable for storing more complex data structures, while Memcached is just a plain key/value store. The simpli
  it to be slightly faster and allows it to scale out if needed, but Redis has more features which you may use in you
- For Memcached AWS provides enhanced SDKs for certain programming languages which implement auto-disco
  available in the normal memcached client libraries.

Back to top ⬆️

### ElastiCache Gotchas and Limitations

- Since in some cases changing the cache clusters may have some restrictions, like for scaling purposes, it may be
  were launched using CloudFormation in a stack that also contains other resources and you really need to chang
  avoid getting your CloudFormation stacks in a non-updateable state, it is recommended to launch ElastiCache c
  resource with similar constraints) in dedicated stacks which can be replaced entirely with new stacks having the

## DynamoDB

### DynamoDB Basics

- 📒 Homepage · Developer guide · FAQ · Pricing
- **DynamoDB** is a NoSQL database with focuses on speed, flexibility, and scalability.
- DynamoDB is priced on a combination of throughput and storage.

Back to top ⬆️

### DynamoDB Alternatives and Lock-in

- ⛓️ Unlike the technologies behind many other Amazon products, DynamoDB is a proprietary AWS product with
  alternative available as an open source project. If you tightly couple your application to its API and featureset, it
  to replace.

- The most commonly used alternative to DynamoDB is Cassandra.

Back to top ⬆️

## DynamoDB Tips

- There is a **local version of DynamoDB** provided for developer use.
- DynamoDB Streams provides an ordered stream of changes to a table. Use it to replicate, back up, or drive even
- DynamoDB can be used as a simple locking service.
- DynamoDB indexing can include **primary keys**, which can either be a single-attribute hash key or a composite h also query non-primary key attributes using **secondary indexes**.
- **Data Types:** DynamoDB supports three data types – **number**, **string**, and **binary** – in both scalar and multi-value support **JSON**.
- As of late 2017, DynamoDB supports both global tables and backup / restore functionality.

Back to top ⬆️

## DynamoDB Gotchas and Limitations

- 🔸 DynamoDB doesn't provide an easy way to bulk-load data (it is possible through Data Pipeline) and this has consequences. Since you need to use the regular service APIs to update existing or create new rows, it is commo destination table's write throughput to speed import. But when the table's write capacity is increased, DynamoD split of the partitions underlying the table, spreading the total table capacity evenly across the new generation c capacity is reduced, the capacity for each partition is also reduced but the total number of partitions is not, leavi partition. This leaves the table in a state where it much easier for hotspots to overwhelm individual partitions.
- 🔸 It is important to make sure that DynamoDB resource limits are compatible with your dataset and workload. maximum size value that can be added to a DynamoDB table is 400 KB (larger items can be stored in S3 and a U
- 🔸 Dealing with **time series data** in DynamoDB can be challenging. A global secondary index together with dow can be a possible solution as explained here.
- 🔸 When setting up fine grained policies for access to DynamoDB tables, be sure to include their secondary ind document as well.

# ECS

## ECS Basics

- 📒 Homepage · Developer guide · FAQ · Pricing
- **ECS** (EC2 Container Service) is a relatively new service (launched end of 2014) that manages clusters of services c
- See the Containers and AWS section for more context on containers.
- ECS is growing in adoption, especially for companies that embrace microservices.
- Deploying Docker directly in EC2 yourself is another common approach to using Docker on AWS. Using ECS is n not (yet) seem to be the predominant way many companies are using Docker on AWS.
- It's also possible to use Elastic Beanstalk with Docker, which is reasonable if you're already using Elastic Beanstal
- Using Docker may change the way your services are deployed within EC2 or Elastic Beanstalk, but it does not rac other services are used.
- ECR (EC2 Container Registry) is Amazon's managed Docker registry service. While simpler than running your ow some features that might be desired by some users:
  - Doesn't support cross-region replication of images.
    - If you want fast fleet-wide pulls of large images, you'll need to push your image into a region-local reg
  - Doesn't support custom domains / certificates.
- A container's health is monitored via CLB or ALB. Those can also be used to address a containerized service. Wh need to handle port contention (i.e. services exposing the same port on the same host) since an ALB's target gro ECS-based services directly.
- The Hitchhikers Guide to AWS ECS and Docker by J. Cole Morrison is an excellent article for Introduction to AWS

Back to top ⬆️

## ECS Tips

- **Log drivers:** ECS supports multiple log drivers (awslogs, splunk, fluentd, syslog, json, ... ). Use `awslogs` for Cloud is made for the logs first). Drivers such as fluentd are not enabled by default. You can, install the agent and enab `ECS_AVAILABLE_LOGGING_DRIVERS='["awslogs","fluentd"]'` to `/etc/ecs/ecs.config` .

- This blog from Convox (and commentary) lists a number of common challenges with ECS as of early 2016.
- It is possible to optimize disk clean up on ECS. By default, the unused containers are deleted after 3 hours and th minutes. These settings can be changed by adding `ECS_ENGINE_TASK_CLEANUP_WAIT_DURATION=10m` and `ECS_IMAGE` `/etc/ecs/ecs.config` . More information on optimizing ECS disk cleanup.

Back to top ⬆️

### ECS Alternatives and Lock-in

- Kubernetes: Extensive container platform. Available as a hosted solution on Google Cloud, AWS, Azure, DigitalO
- Nomad: Orchestrator/Scheduler, tightly integrated in the HashiCorp stack (Consul, Vault, etc).

🚧 *Please help expand this incomplete section.*

## EKS

### EKS Basics

- 📒 Homepage · User guide · FAQ · Pricing
- EKS (Elastic Kubernetes Service) is a new service (launched June 2018) that provides managed Kubernetes Maste to deploy K8s Services and Pods on top of EC2 based Kubernetes nodes.
- See the Containers and AWS section for more context on containers.
- EKS is AWS's solution to hosting Kubernetes natively on AWS. It is not a replacement for ECS directly but is in res dominance of Kubernetes.
- EKS does not launch EC2 nodes and would have to be configured and setup either manually or via Cloudformati solution)
- EKS management is done through a utility called kubectl, and with Kube configuration files. These files will need with the K8s Master with a certificate and URL. The AWS CLI can autogenerate the configuration file that kubectl communicating with the cluster.[1]

- EKS authentication is integrated with IAM roles/permissions. The AWS CLI has an integrated sub-command for g tokens.[2] This was formerly done via a custom plugin for kubectl called aws-iam-authenticator (formerly heptio-a
- EKS provides Calico from Tigera for securing workloads within a cluster using Kubernetes network policy.

Back to top ⬆️

## EKS Tips

- Multiple clusters can be supported by using different kubeconfig files.
- AWS has a Kubernetes Quickstart developed in collaboration with Heptio.

Back to top ⬆️

## EKS Alternatives and Lock-in

- ECS: Amazon's native Container Scheduled platform released in 2014. If you don't utilise containers today and a ECS is an excellent product.
- Kubernetes: Extensive container platform. Available as a hosted solution on Google Cloud, AWS, Digital Ocean a
- Nomad: Orchestrator/Scheduler, tightly integrated in the HashiCorp stack (Consul, Vault, etc).

Back to top ⬆️

## EKS Gotchas and Limitations

- Pods and Service configurations can rapidly consume IP addresses inside a VPC. Proper care and maintenance sh IP exhaustion does not occur.
- There is currently no integrated monitoring in CloudWatch for EKS pods or services, you will need to deploy a m supports Kubernetes such as Prometheus.
- Autoscaling based off CPU/Memory of a node is limited as you will not be aware of pending Services/Pods that autoscaler can be useful for scaling based on Node resource usage and unschedulable Pods.
- Prometheus is a very popular monitoring solution for K8s, metrics and alerts can be used to send events to Lam to take autoscaling actions.

Back to top ⬆️

### Footnotes

1: https://docs.aws.amazon.com/eks/latest/userguide/create-kubeconfig.html
2: https://aws.amazon.com/about-aws/whats-new/2019/05/amazon-eks-simplifies-kubernetes-cluster-authentication

## Fargate

### Fargate Basics

- 📒 Homepage · FAQ · Pricing
- **Fargate** allows you to manage and deploy containers without having to worry about running the underlying com
- Fargate serves as a new backend (in addition to the legacy EC2 backend) on which ECS and EKS tasks can be run
- Fargate and EC2 backends are called "Launch Types"
- Fargate allows you to treat containers as fundamental building blocks of your infrastructure

Back to top ⬆️

### Fargate Tips

- Fargate follows a similar mindset to Lambda, which lets you focus on applications, instead of dealing with under
- Fargate is supported by CloudFormation, aws-cli and ecs-cli
- Fargate tasks can be launched alongside tasks that use EC2 Launch Type
- 💸 Before creating a large Fargate deployment, make sure to estimate costs and compare them against alternati traditional EC2 deployment - Fargate prices can be several times those of equivalently-sized EC2 instances. To ev based on potential costs, refer to pricing for EC2 and Fargate.

Back to top ⬆️

### Fargate Alternatives and Lock-in

- Azure Container Instances: Available on Microsoft Azure in preview version, allows to run applications in cont
  manage virtual machines

Back to top ⬆️

### Fargate Gotchas and Limitations

- The smallest resource values that can be configured for an ECS Task that uses Fargate is 0.25 vCPU and 0.5 GB o
- Task storage is ephemeral. After a Fargate task stops, the storage is deleted.

## Lambda

### Lambda Basics

- 📒 Homepage · Developer guide · FAQ · Pricing
- **Lambda** is AWS' serverless compute offering, allowing users to define Lambda functions in a selection of runtim
  a variety of triggers, including SNS notifications and API Gateway invocations. Lambda is the key service that ena
  architecture on AWS, alongside AWS API Gateway, AWS Batch, and AWS DynamoDB.

Back to top ⬆️

### Lambda Tips

- The idea behind 'serverless' is that users don't manage provisioning, scaling, or maintenance of the physical mac
  application code. With Lambda, the machine that actually executes the user-defined function is abstracted as a '
  a Lambda function, users are able to declare the amount of memory available to the function, which directly affe
  specification of the Lambda container.
- Changing the amount of memory available to your Lambda functions also affects the amount of CPU power ava
- While AWS does not offer hard guarantees around container reuse, in general it can be expected that an unalter
  reuse a warm (previously used) container if called shortly after another invocation. Users can use this as a way to
  by smartly caching application data on initialization.

- A Lambda that hasn't been invoked in some time may not have any warm containers left. In this case, the Lambc and initialize the Lambda code in a 'cold start' scenario, which can add significant latency to Lambda invocations performance has improved significantly over the 2018-2019 timeframe and is now typically in the range of 200-5 function depending on the language runtime.
- Lambda functions running insides of VPCs have also seen recent improvements to cold start times. Previously th would have cold starts of ~15 seconds; now those same functions cold start in < 1 second.
- There are a few strategies to avoiding or mitigating cold starts. Provisioned concurrency was announced at re:in effective means to eliminating cold starts. Other techniques include keeping containers warm by periodic trigge lightweight runtimes such as Node as opposed to Java.
- Lambda is integrated with AWS CloudWatch and provides a logger at runtime that publishes CloudWatch events
- Lambda offers out-of-the-box opt-in support for AWS X-Ray. X-Ray can help users diagnose Lambda issues by c of their Lambda's execution flow. This is especially useful when investigating issues calling other AWS services as detailed and easy-to-parse visualization of the call graph.
- Using timed CloudWatch events, users can use Lambda to run periodic jobs in a cron-like manner.
- Events sent to Lambda that fail processing can be managed using a Dead Letter Queue (DLQ) in SQS.
- More on serverless:
  - Mike Roberts's thoughts on martinfowler.com.
  - AWS Serverless Application Model (SAM), a simplification built on top of CloudFormation that can help to d serverless applications using Lambda.
  - Serverless, one of the most popular frameworks for building serverless applications using AWS Lambda and options.
  - Other helpful frameworks.

Back to top ⬆️

## Lambda Alternatives and Lock-in

- 🚪 Other clouds offer similar services with different names, including Google Cloud Functions, Azure Functions, if you are running Kubernetes another Lambda alternative is OpenFaaS

Back to top ⬆️

## Lambda Gotchas and Limitations

- 🔸 Testing Lambdas, locally and remotely, can be a challenge. Several tools are available to make this easier, incl
  supported SAM Local.
- 🔸 Managing lots of Lambda functions is a workflow challenge, and tooling to manage Lambda deployments is
- 🔸 AWS' official workflow around managing function versioning and aliases is painful. One option is to avoid La
  abstracting your deployment workflow outside of Lambda. One way this can be accomplished is by deploying yo
  successive stages, with a distinct AWS account per stage, where each account only needs to be aware of the late
  and updates are handled by external tooling.
- 🔸 While adding/removing S3 buckets as triggers for Lambda function, this error may occur: "There was an error
  Configuration is ambiguously defined. Cannot have overlapping suffixes in two rules if the prefixes are overlapp
  type." In this case, you can manually remove the Lambda event in the "Events" tab in the "Properties" section of
- 🔸 Managing the size of your deployment artifact can be a challenge, especially if using Java. Options to mitigat
  and loading dependencies at runtime into /tmp.
- When using DynamoDB as a trigger for your Lambda functions, this error may occur: "PROBLEM: internal Lambd
  Lambda customer support." This usually just means that Lambda can't detect anything in the DynamoDB stream
  the issue persists, deleting and recreating your trigger may help.
- 🔸 If your lambda needs access to resources in a VPC (for example ElastiCache or RDS), it will need to be deploy
  increase cold-start times as an Elastic Network Interface (ENI) will have to be registered within the VPC for each
  also has a relatively low initial limit (350) on the number ENI's that can be created within an VPC, however this ca
  1000s if a good case is made to AWS support.
- 🔸 If your lambda is in a VPC and needs access to resources outside VPC such outbound internet access, it will n
  associated even if it is in public subnet. Lambdas won't be assigned a public IP irrespective of subnet. If it is acce
  example S3), you may use VPC endpoints.
- 🔸 Lambda has several **resource limits** as of 2017-06:
  - A **6MB** request or response payload size.
  - A **50 MB** limit on the compressed .zip/.jar file deployment package size.
  - A **250 MB** limit on the code/dependencies in the package before compression.

- A **500 MB** limit on local storage in /tmp.

Back to top ⬆️

### Lambda Code Samples

- Fan-out is an example of using Lambda to "fan-out" or copy data from one service, in this case Kinesis, to multip Destinations for fan-out data in the sample include IoT, SQS and more.
- This AWS limit monitor using Lambdas shows use of multiple Lambdas for monitoring.
- This Lambda ECS Worker Pattern shows use of Lambda in a workflow where data from S3 is picked up by the La then sent to ECS for more processing.
- The Secure Pet Store is a sample Java application which uses Lambda and API Gateway with Cognito (for user id
- aws-lambda-list is a list of "hopefully useful AWS lambdas and lambda-related resources". Quite a few code sam guaranteed tested. Caveat Emptor.

🚧 *Please help expand this incomplete section.*

## API Gateway

### API Gateway Basics

- 📒 Homepage · Developer guide · FAQ · Pricing
- **API Gateway** provides a scalable, secured front-end for service APIs, and can work with Lambda, Elastic Beanstal
- It allows "serverless" deployment of applications built with Lambda.
- 🔸 Switching over deployments after upgrades can be tricky. There are no built-in mechanisms to have a single one API gateway to another one. So it may be necessary to build an additional layer in front (even another API G migration from one deployment to another.

Back to top ⬆️

### API Gateway Alternatives and Lock-In

- Kong is an open-source, on-premises API and microservices gateway built on nginx with Lua. Kong is extensible
- Tyk is an open-source API gateway implemented in Go and available in the cloud, on-premises or hybrid.

Back to top ⬆️

## API Gateway Tips

- 🔹 Prior to 2016-11, you could only send and receive plain text data (so people would base64-encode binary da
supported.
- API Gateway supports the OpenApi specification (aka Swagger). This allows you to describe your API in a langua
various tools to generate code supporting your API.
- Generating clients is extremely easy, either through the AWS console or using the get-sdk API.
- API Gateway integrates with CloudWatch out-of-the-box, allowing for easy logging of requests and responses.
  - Note that if your request or response are too large, CloudWatch will truncate the log. For full request/reply
  so in your integration (e.g. Lambda).
  - A good practice when calling API Gateway APIs is to log the request ID on the client. You can later refer to t
  CloudWatch for easier tracing and debugging.
- There are multiple ways to secure your API, including built-in support for AWS Cognito. For most use-cases, Cog
simplest way to authenticate users.
  - Although you can roll your own solution using a custom authorizer, which is basically a Lambda you define
  request is acceptable or not.
- While API Gateway lends itself well to REST-style development, it's perfectly reasonable to implement an RPC-st
well. Depending on your use-case, this can often lead to a much simpler API structure and smoother client expe
  - RPC-style APIs are particularly useful when designing services that sit deeper in the stack and don't serve co

Back to top ⬆️

## API Gateway Gotchas and Limitations

- 🔸 API Gateway only supports encrypted (https) endpoints, and does not support unencrypted HTTP. (This is pro

- 🔸 API Gateway doesn't support multi-region deployments for high availability. It is a service that is deployed in with a global endpoint that is served from AWS edge locations (similar to a CloudFront distribution). You cannot Gateways with the same hostname in different AWS regions and use Route 53 to distribute the traffic. More in th
- 🔸 Integration timeout: All of the various integration types (eg: Lambda, HTTP) for API Gateway have timeouts, a some limits, these timeouts can't be increased.
- 🔸 API Gateway returns a 504 status code for any network or low level transport related issue. When this happen in the CloudWatch logs for the request that includes the message: `Execution failed due to an internal error` this error is that even though your backend server is up and running, it may be doing something outside of the sending well-formed chunked messages). You can test by hitting your backend directly with the `curl --raw -S` and seeing if it complains.
- 🔸 AWS X-Ray support exists but cumbersome to use. If you have other AWS services calling API Gateway, your there. API Gateway will also not appear as a node in your service map. More here.
- 🔸 Be careful using the export feature. The resulting Swagger template is often incomplete and doesn't integrate extensions for things such as CORS.
- 🔸 Many changes to API Gateway resources need to be 'deployed' via console or API call. Unfortunately, API Gat notifying the user when changes are staged for deployment and what changes require deployment. If you've ch your API and it's not taking effect, there's a decent chance you just need to deploy it.
  - In particular, when deploying an API Gateway as part of a CloudFormation stack, changes will not automatic deployment resource itself was changed. You can change work around this by always changing the deploym CloudFormation update, or running a custom resource that ensures the deployment is made.
  - Alternatively, by using the Serverless Application Model definition for an API Gateway resource, you can alw deployed on a stack update since SAM will generate a new deployment every time.
- 🔸 API Gateway does not support nested query parameters on method requests.
- 🔸 API Gateway limits number of resources to 300, as described here. This is something to be considered when Gateway as a platform where your team/organization deploys to the same API Gateway.

🚧 *Please help expand this incomplete section.*

# Step Functions

## Step Functions Basics

- 📒 Homepage · Developer guide · FAQ · Pricing
- **Step Functions** is AWS' way to create state machines that manage a serverless workflow.

Back to top ⬆️

## Step Functions Tips

- A variety of structures are supported including branching, parallel operations and waits
- Tasks represent the real work nodes and are frequently Lambda functions, but can be Activities which are extern
  implemented any way you like.
- State machines have data that "flows" through the steps and can be modified and added to as the state machin
- It's best if your tasks are idempotent, in part because you may want to re-run the state machine with the same i
  debugging
- The AWS Console facilitates your examining the execution state at various steps.
  - The console lets you do this with a few steps:
    - select the "input" tab from the failed execution
    - copy the input data (JSON)
    - select the state machine name in the breadcrumbs
    - start a new execution, pasting the input data you copied previously

Back to top ⬆️

## Step Functions Gotchas and Limitations

- Step Functions are free tier eligible up to an initial 4000 transitions per month. Thereafter, the charge is $0.025 p
- You can have many, simultaneous, executions, but be aware of lambda throttling limits. This has been per-accou
  became settable per-lambda.

- Step Function executions are limited to 25,000 events. Each step creates multiple events. This means that iteratin limited to an iteration count of around 3000 before needing to continue as a new execution.

## Route 53

### Route 53 Basics

- 📒 Homepage · Developer guide · FAQ · Pricing
- **Route 53** is AWS' DNS service.

Back to top ⬆️

### Route 53 Alternatives and Lock-In

- Historically, AWS was slow to penetrate the DNS market (as it is often driven by perceived reliability and long-te but Route 53 has matured and is becoming the standard option for many companies. Route 53 is cheap by histo has a fairly large global network with geographic DNS and other formerly "premium" features. It's convenient if
- ⛓️ Generally you don't get locked into a DNS provider for simple use cases, but increasingly become tied in onc like geographic routing or Route 53's alias records.
- 🚪 Many alternative DNS providers exist, ranging from long-standing premium brands like UltraDNS and Dyn to modestly priced brands like DNSMadeEasy. Most DNS experts will tell you that the market is opaque enough th performance don't really correlate well with price.
- ⏱️ Route 53 is usually somewhere in the middle of the pack on performance tests, e.g. the SolveDNS reports.

Back to top ⬆️

### Route 53 Tips

- 🔹 Know about Route 53's "alias" records:
  - Route 53 supports all the standard DNS record types, but note that **alias resource record sets** are not stand specific Route 53 feature. (It's available from other DNS providers too, but each provider has a different nam

    - Aliases are like an internal name (a bit like a CNAME) that is resolved internally on the server side. For exam
      have a CNAME to the DNS name of a CLB or ALB, but it's often better to make an alias to the same load bal
      same, but in the latter case, externally, all a client sees is the target the record points to.
    - It's often wise to use alias record as an alternative to CNAMEs, since they can be updated instantly with an A
      about DNS propagation.
    - You can use them for CLBs/ALBs or any other resource where AWS supports it.
    - Somewhat confusingly, you can have CNAME and A aliases, depending on the type of the target.
    - Because aliases are extensions to regular DNS records, if exported, the output zone file will have additional
      in it.
- Latency-based routing allows users around the globe to be automatically directed to the nearest AWS region w
  that latency is reduced.
- Understand that domain registration and DNS management (hosted zones) are two separate Route 53 services.
  domain, Route 53 automatically assigns four name servers to it (e.g. ns-2.awsdns-00.com). Route 53 also offers t
  hosted zone for DNS management, but you are not required do your DNS management in the same account or
  need to create an NS record pointing to the servers assigned to your domain in Route 53.
- One use case would be to put your domain registration (very mission critical) in a bastion account while managi
  another account which is accessible by your applications.

Back to top ⬆️

### Route 53 Gotchas and Limitations

- 🔸 Private Hosted Zone will only respond to DNS queries that originate from within a VPC. As a result Route53 w
  made via a VPN or Direct connect. To get around this you will need to implement Hybrid Cloud DNS Solutions o
  provided IP addresses to query the hosted zone.

## CloudFormation

### CloudFormation Basics

- 📒 Homepage · Developer guide · FAQ · Pricing at no additional charge
- **CloudFormation** allows you to manage sets of resources from other AWS services grouped into **stacks**. CloudFo define these stacks in a template using JSON or YAML. CloudFormation is one of the major services underpinnin code capabilities and is crucial in enabling repeatable and consistent deployments of infrastructure.
- 💸 CloudFormation itself has no additional charge; you pay for the underlying resources.

Back to top ⬆️

## CloudFormation Alternatives and Lock-In

- HashiCorp's [Terraform](https://www. /intro/vs/cloudformation.html) is a third-party alternative that can support platforms/providers including Azure and OpenStack.
- 🔸 Some AWS features may not be available in Terraform (e.g. multi-AZ ElastiCache using Redis), and you may h CloudFormation templates.
- Pulumi enables teams to define and deliver Cloud Native Infrastructure as Code on any cloud, with any language serverless to Kubernetes to infrastructure.

Back to top ⬆️

## CloudFormation Tips

- Validate your stack in a different AWS account! CloudFormation truly shines when making multiple deployments different accounts and regions. A common practice is to deploy stacks in successive stages ending in a productio
- Avoid potentially time-consuming syntax errors from eating into your deployment time by running `validate-te`
- CloudFormation is sometimes slow to update what resources (and new features on old services) a user is able to you need to deploy a resource or feature that isn't supported by the template, CloudFormation allows running a Lambda) on a stack create or update via custom resources.
- Custom resources make CloudFormation into a truly powerful tool, as you can do all sorts of neat things quite e initial configuration of Dynamo tables or S3 buckets, cleaning up old CloudWatch logs, etc.
  - For writing Custom Resources in Javascript, AWS provides a good reference in the documentation.
- CloudFormation offers a visual template designer that can be useful when getting up to speed with the template

- By using StackSets, users can define and deploy an entire production application consisting of multiple stacks (o single CloudFormation template.
- If you're developing a serverless application (i.e., using Lambda, API Gateway) CloudFormation offers a simplifie SAM.
- ❗ Use a restrictive stack policy! Without one, you can inadvertently delete live production resources, probably c
- ❗ Turn on termination protection on all of your stacks to avoid costly accidents!
- The CloudFormation template reference is indispensable when discovering what is and isn't possible in a CloudF
- Troposphere is a Python library that makes it much easier to create CloudFormation templates.
  - Currently supports AWS and OpenStack resource types.
  - Troposphere attempts to support all resources types that can be described in CloudFormation templates.
  - Built in error checking.
  - A recommended soft dependency is awacs, which allows you to generate AWS access policy in JSON by wri
- stacker is a Python application that makes it easy to define, configure, orchestrate and manage dependencies fo across multiple user-defined environments.
- If you are building different stacks with similar layers, it may be useful to build separate templates for each layer AWS::CloudFormation::Stack.
- 🔸 Avoid hardcoding resource parameters that can potentially change. Use stack parameters as much as you ca parameter values.
- 🔹 Until 2016, CloudFormation used only an awkward JSON format that makes both reading and debugging diff typically involved building additional tooling, including converting it to YAML, but now this is supported directly
- Wherever possible, export relevant physical IDs from your Stacks by defining Outputs in your CloudFormation Te actual names assigned to the resources being created. Outputs can be returned from `DescribeStack` API calls, a Stacks as part of the recent addition of cross-stack references. -Note that importing outputs in a stack from ano dependency that is tracked by CloudFormation. You will not be able to delete the stack with the outputs until th stacks.
- CloudFormation can be set up to send SNS notifications upon state changes, enabling programmatic handling c fail to build, or simple email alerts so the appropriate people are informed.
- CloudFormation allows the use of **conditionals** when creating a stack.

    - One common way to leverage this capability is in support of multi-environment CloudFormation templates use 'if-else' statements on the value of a parameter passed in (e.g. "env"), environment-specific values for th SecurityGroup IDs, and AMI names can be passed into reusable generic templates.
- **Version control your CloudFormation templates!** In the Cloud, an application is the combination of the code w it runs on. By version controlling **both**, it is easy to roll back to known good states.
- Avoid naming your resources explicitly (e.g. DynamoDB tables). When deploying multiple stacks to the same AW can come into conflict, potentially slowing down your testing. Prefer using resource references instead.
- For things that shouldn't ever be deleted, you can set an explicit DeletionPolicy on the resource that will prevent deleted even if the CloudFormation stack itself is deleted. This is useful for anything that can maintain expensive DynamoDB tables, and things that are exposed to the outside world, such as API Gateway APIs.

Back to top ⬆️

## CloudFormation Gotchas and Limitations

- 🔸 A given CloudFormation stack can end up in a wide variety of states. Error reporting is generally weak, and of observe-tweak-redeploy cycles are needed to get a working template. The internal state machine for all the vary opaque.
- 🔸 Some cross-region operations are not possible in CloudFormation without using a custom resource, such as subscriptions.
- 🔸 While having hand-made resources live alongside CloudFormation-created resources is inadvisable, it's some possible, leave ALL resource management up to a CloudFormation template and only provide read-only access t
- ❗ Modifications to stack resources made outside CloudFormation can potentially lead to stacks stuck in UPDAT mode. Stacks in this state can be recovered using the continue-update-rollback command. This command can b in the CLI. The --resources-to-skip parameter usable in the CLI can be useful if the continue-update-rollback com Drift Detection can be used to detect outside changes made to stack.
- 🔸 CloudFormation is useful but complex and with a variety of pain points. Many companies find alternate solut use it, but only with significant additional tooling.
- 🔸 CloudFormation can be very slow, especially for items like CloudFront distributions and Route53 CNAME entr
- 🔸 It's hard to assemble good CloudFormation configurations from existing state. AWS does offer a trick to do t

    - CloudFormer also hasn't been updated in ages (as of Oct 2017), doesn't support templatizing many new ser
      define even existing services that have since been updated. For example, Dynamo tables defined through C
      TTL definitions or auto-scaling configuration. There is a third-party version of the tool with more supported
- 🔸 Many users don't use CloudFormation at all because of its limitations, or because they find other solutions pr
  other ways to accomplish the same goals, such as local scripts (Boto, Bash, Ansible, etc.) you manage yourself th
  Docker-based solutions (Convox, etc.).
- 🔸 Deploying large stacks (i.e., many resources) can be problematic due to unintuitive API limits. For instance, AP
  `CreateDeployment` API has a default limit of 3 requests per minute as of 1/12/2018. This limit is readily exceeded
  CloudFormation stacks. Creating CW alarms is another commonly seen limit ( `PutMetricAlarm` , 3 tps as of 1/12/2
  creating many autoscaling policies for DynamoDB. One way to work around this limit is to include CloudFormati
  artificially chain resource creation.
- 🔸 Creating/deleting stacks can be a little less clean than ideal. Some resources will leave behind traces in your A
  deletion. E.g., Lambda will leave behind CloudWatch log groups that never expire.

## VPCs, Network Security, and Security Groups

### VPC Basics

- 📒 Homepage · User guide · FAQ · Security groups · Pricing
- **VPC** (Virtual Private Cloud) is the virtualized networking layer of your AWS systems.
- Most AWS users should have a basic understanding of VPC concepts, but few need to get into all the details. VP
  trivial or extremely complex, depending on the extent of your network and security needs.
- All modern AWS accounts (those created after 2013-12-04) are "EC2-VPC" accounts that support VPCs, and all in
  VPC. Older accounts may still be using "EC2-Classic" mode. Some features don't work without VPCs, so you prob

Back to top ⬆️

### VPC and Network Security Tips

- ❗ **Security groups** are your first line of defense for your servers. Be extremely restrictive of what ports are open connections. In general, if you use CLBs, ALBs or other load balancing, the only ports that need to be open to inc port 22 and whatever port your application uses. Security groups access policy is 'deny by default'.

- **Port hygiene:** A good habit is to pick unique ports within an unusual range for each different kind of production web frontend might use 3010, your backend services 3020 and 3021, and your Postgres instances the usual 5432 fine-grained security groups for each set of servers. This makes you disciplined about listing out your services, b For example, should you accidentally have an extra Apache server running on the default port 80 on a backend s exposed.

- **Migrating from Classic**: For migrating from older EC2-Classic deployments to modern EC2-VPC setup, this articl
    - You can migrate Elastic IPs between EC2-Classic and EC2-VPC.

- For basic AWS use, one default VPC may be sufficient. But as you scale up, you should consider mapping out net thoroughly. A good overview of best practices is here.

- Consider controlling access to you private AWS resources through a VPN.
    - You get better visibility into and control of connection and connection attempts.
    - You expose a smaller surface area for attack compared to exposing separate (potentially authenticated) serv internet.
        - e.g. A bug in the YAML parser used by the Ruby on Rails admin site is much less serious when the adm private network and accessed through VPN.
    - Another common pattern (especially as deployments get larger, security or regulatory requirements get mo increase) is to provide a bastion host behind a VPN through which all SSH connections need to transit.
    - For a cheap VPN to access private AWS resources, consider using a point-to-site software VPN such as Ope installed using the official AMI, though you are limited to 2 concurrent users on the free license, or it can be openvpn package on linux. The linux package allows for unlimited concurrent users but the installation is le OpenVPN installer script can help you install it and add client keys easily.

- 🔹 Consider using other security groups as sources for security group rules instead of using CIDRs — that way, a security group and only hosts in that security group are allowed access. This is a much more dynamic and secure group rules.

- **VPC Flow Logs** allow you to monitor the network traffic to, from, and within your VPC. Logs are stored in CloudW can be used for security monitoring (with third party tools), performance evaluation, and forensic investigation.

  - See the VPC Flow Logs User Guide for basic information.
  - See the flowlogs-reader CLI tool and Python library to retrieve and work with VPC Flow Logs.

- **IPv6** is available in VPC. Along with this announcement came the introduction of the Egress-Only Internet Gatew would use NAT Gateways to enable egress-only traffic for their VPC in IPv4, one can use an Egress-Only Internet purpose in IPv6.

- Amazon provides an IPv6 CIDR block for your VPC at your request - at present you cannot implement your own to own one already.

- New and existing VPCs can both use IPv6. Existing VPCs will need to be configured to have an IPv6 CIDR block a as new VPCs do.

Back to top ⬆️

## PrivateLink

- 📒 Homepage · User Guide · Pricing
- One of the uses for Private link is Interface VPC Endpoints deploys an ENI into your VPC and subnets which allow AWS API's as if the were accessible locally in your VPC without having to go out to the internet.
- Another use case would be to expose a service of your own to other accounts in AWS through a VPC Endpoint S

Back to top ⬆️

## VPC and Network Security Gotchas and Limitations

- 🔸 VPCs are tied to one Region in one Account. Subnets are tied to one VPC and limited to one Availability Zone

- 🔸 Security groups are tied to one VPC. If you are utilizing infrastructure in multiple VPCs you should make sure configuration/deployment tools take that into account.
- 🔸 VPC Endpoints are currently only available for S3 and DynamoDB. If you have a security requirement to lockd your VPC you may want to use DNS filtering to control outbound traffic to other services.
- ❗ Be careful when choosing your VPC IP CIDR block: If you are going to need to make use of ClassicLink, make range doesn't overlap with that of EC2 Classic.
- ❗ If you are going to peer VPCs, carefully consider the cost of data transfer between VPCs, since for some work can be prohibitively expensive.
- ❗ New RDS instances require a subnet group within your VPC. If you're using the default VPC this isn't a concer for each availability zone in your region. However, if you're creating your own VPC and plan on using RDS, make subnets within the VPC to act as the subnet group.
- ❗ If you delete the default VPC, you can recreate it via the CLI or the console.
- ❗ Be careful with VPC VPN credentials! If lost or compromised, the VPN endpoint must be deleted and recreate Replacing Compromised Credentials.
- ❗ Security Groups and Route Tables apply entries separately for IPv4 and IPv6, so one must ensure they add en accordingly.
- 💸 Managed NAT gateways are a convenient alternative to manually managing NAT instances, but they do come Consider alternatives if you're transferring many terabytes from private subnets to the internet. If you transfer te from EC2 instances in private subnets to S3, avoid the NAT gateway data processing charge by setting up a Gate and route the traffic to/from S3 through the VPC endpoints instead of going through the NAT gateways.

## KMS

### KMS Basics

- 📒 Homepage · Developer guide · FAQ · Pricing
- **KMS** (Key Management Service) is a secure service for creating, storing and auditing usage of cryptographic key
- **Service integration:** KMS integrates with other AWS services: EBS, Elastic Transcoder, EMR, Redshift, RDS, SES, S3 Workspaces.

- **Encryption APIs:** The Encrypt and Decrypt API allow you to encrypt and decrypt data on the KMS service side, n key contents.
- **Data keys:** The GenerateDataKey API generates a new key off of a master key. The data key contents are expose to encrypt and decrypt any size of data in your application layer. KMS does not store, manage or track data keys this in your application.
- 🔹 **Auditing:** Turn on CloudTrail to audit all KMS API events.
- **Access:** Use key policies and IAM policies to grant different levels of KMS access. For example, you create an IAM user to encrypt and decrypt with a specific key.

Back to top ⬆️

## KMS Tips

- 🔹 It's very common for companies to manage keys completely via home-grown mechanisms, but it's far prefera as KMS from the beginning, as it encourages more secure design and improves policies and processes around m
- A good motivation and overview is in this AWS presentation.
- The cryptographic details are in this AWS whitepaper.
- This blog from Convox demonstrates why and how to use KMS for encryption at rest.

Back to top ⬆️

## KMS Gotchas and Limitations

- 🔸 The Encrypt API only works with < 4KB of data. Larger data requires generating and managing a data key in y
- 🔸 KMS audit events are not available in the CloudTrail Lookup Events API. You need to look find them in the raw CloudTrail saves in S3.
- 🔸 In order to encrypt a multi-part upload to S3, the KMS Key Policy needs to allow "kms:Decrypt" and "kms:Ger to "kms:Encrypt", otherwise the upload will fail with an "AccessDenied" error.
- 🔸 KMS keys are region specific — they are stored and can only be used in the region in which they are created. to other regions.

- 🔸 KMS keys have a key policy that must grant access to something to manage the key. If you don't grant anyth creation, then you have to reach out to support to have the key policy reset Reduce the Risk of the Key Becomin
- 🔸 If you use a key policy to grant access to IAM roles or users and then delete the user/role, recreating the user permission to the key again.

## CloudFront

### CloudFront Basics

- 📒 Homepage · Developer guide · FAQ · Pricing
- **CloudFront** is AWS' content delivery network (CDN).
- Its primary use is improving latency for end users through accessing cacheable content by hosting it at over 60

Back to top ⬆️

### CloudFront Alternatives and Lock-in

- 🚪 CDNs are a highly fragmented market. CloudFront has grown to be a leader, but there are many alternatives specific needs.

Back to top ⬆️

### CloudFront Tips

- 🐥 **IPv6** is supported. This is a configurable setting, and is enabled by default on new CloudFront distributions. I use of WAF with CloudFront.
- 🐥 **HTTP/2** is now supported! Clients must support TLS 1.2 and SNI.
- While the most common use is for users to browse and download content (GET or HEAD methods) requests, Clo (since 2013) uploaded data (POST, PUT, DELETE, OPTIONS, and PATCH).
  - You must enable this by specifying the allowed HTTP methods when you create the distribution.
  - Interestingly, the cost of accepting (uploaded) data is usually less than for sending (downloaded) data.

- In its basic version, CloudFront supports SSL via the SNI extension to TLS, which is supported by all modern web support older browsers, you need to pay a few hundred dollars a month for dedicated IPs.
  - 💸 ⏱️ Consider invalidation needs carefully. CloudFront does support invalidation of objects from edge loca takes many minutes to propagate to edge locations, and costs $0.005 per request after the first 1000 reque support this better.)
- Everyone should use TLS nowadays if possible. Ilya Grigorik's table offers a good summary of features regarding of CloudFront.
- An alternative to invalidation that is often easier to manage, and instant, is to configure the distribution to cache then append unique query strings with versions onto assets that are updated frequently.
- ⏱️ For good web performance, it is recommended to enable compression on CloudFront distributions if the orig that does not already compress.

Back to top ⬆️

### CloudFront Gotchas and Limitations

- 🔸 If using S3 as a backing store, remember that the endpoints for website hosting and for general S3 are differe "bucketname.s3.amazonaws.com" is a standard S3 serving endpoint, but to have redirect and error page suppor website hosting endpoint listed for that bucket, e.g. "bucketname.s3-website-us-east-1.amazonaws.com" (or the
- 🔸 By default, CloudFront will not forward HTTP Host: headers through to your origin servers. This can be proble run multiple sites switched with host headers. You can enable host header forwarding in the default cache behav
- 🔸 4096-bit SSL certificates: CloudFront do not support 4096-bit SSL certificates as of late 2016. If you are using certificate, you'll need to make sure it's 2048 bits. See ongoing discussion.
- Although connections from clients to CloudFront edge servers can make use of IPv6, connections to the origin s IPv4.

## DirectConnect

### DirectConnect Basics

- 📒 Homepage · User guide · FAQ · Pricing
- **Direct Connect** is a private, dedicated connection from your network(s) to AWS.

Back to top ⬆️

### DirectConnect Tips

- If your data center has a partnering relationship with AWS, setup is streamlined.
- Use for more consistent predictable network performance guarantees (**1 Gbps** or **10 Gbps** per link).
- Use to peer your colocation, corporate, or physical datacenter network with your VPC(s).
  - Example: Extend corporate LDAP and/or Kerberos to EC2 instances running in a VPC.
  - Example: Make services that are hosted outside of AWS for financial, regulatory, or legacy reasons callable f

## Redshift

### Redshift Basics

- 📒 Homepage · Developer guide · FAQ · Pricing
- **Redshift** is AWS' managed data warehouse solution, which is massively parallel, scalable, and columnar. It is very using ParAccel technology and exposes Postgres-compatible interfaces.

Back to top ⬆️

### Redshift Alternatives and Lock-in

- ⛓️ 🚪 Whatever data warehouse you select, your business will likely be locked in for a long time. Also (and not c warehouse market is highly fragmented. Selecting a data warehouse is a choice to be made carefully, with resea market landscape and what business intelligence tools you'll be using.

Back to top ⬆️

### Redshift Tips

- Although Redshift is mostly Postgres-compatible, its SQL dialect and performance profile are different.
- Redshift supports only 12 primitive data types. (List of unsupported Postgres types)
- It has a leader node and computation nodes (the leader node distributes queries to the computation ones). Not be executed only on the lead node.
- 🔹 Make sure to create a new cluster parameter group and option group for your database since the default par allow dynamic configuration changes.
- Major third-party BI tools support Redshift integration (see Quora).
- Top 10 Performance Tuning Techniques for Amazon Redshift provides an excellent list of performance tuning tec
- Amazon Redshift Utils contains useful utilities, scripts and views to simplify Redshift ops.
- VACUUM regularly following a significant number of deletes or updates to reclaim space and improve query per
- Avoid performing blanket VACUUM or ANALYZE operations at a cluster level. The checks on each table to deter ANALYZE action needs to be taken is wasteful. Only perform ANALYZE and VACUUM commands on the objects Analyze & Vacuum Schema Utility to perform this work. The SQL to determine whether a table needs to be VAC be found in the Schema Utility README if you wish to create your own maintenance process.
- Redshift provides various column compression options to optimize the stored data size. AWS strongly encourag compression at the COPY stage, when Redshift uses a sample of the data being ingested to analyze the column However, automatic compression can only be applied to an empty table with no data. Therefore, make sure the enough to provide Redshift with a representative sample of the data (the default sample size is 100,000 rows).
- Redshift uses columnar storage, hence it does not have indexing capabilities. You can, however, use distribution improve performance. Redshift has two types of sort keys: compounding sort key and interleaved sort key.
- A compound sort key is made up of all columns listed in the sort key definition. It is most useful when you have using the prefix of the sortkey.
- An interleaved sort key on the other hand gives equal weight to each column or a subset of columns in the sort ahead of time which column(s) you want to choose for sorting and filtering, this is a much better choice than the example using interleaved sort key.
- 🔸 ⏱️ **Distribution strategies:** Since data in Redshift is physically distributed among nodes, choosing the right d distribution style is crucial for adequate query performance. There are three possible distribution style settings – or **ALL**. Use KEY to collocate join key columns for tables which are joined in queries. Use ALL to place the data in cluster nodes.

Back to top ⬆️

## Redshift Gotchas and Limitations

- ❗ ⏱️ While Redshift can handle heavy queries well, it does not scale horizontally, i.e. does not handle multiple c
  Therefore, if you expect a high parallel load, consider replicating or (if possible) sharding your data across multip
- 🔸 The leader node, which manages communications with client programs and all communication with comput
  of failure.
- ⏱️ Although most Redshift queries parallelize well at the compute node level, certain stages are executed on the
  become the bottleneck.
- 🔹 Redshift data commit transactions are very expensive and serialized at the cluster level. Therefore, consider g
  commands (COPY/INSERT/UPDATE) commands into a single transaction whenever possible.
- 🔹 Redshift does not support multi-AZ deployments. Building multi-AZ clusters is not trivial. Here is an example
- 🔸 Beware of storing multiple small tables in Redshift. The way Redshift tables are laid out on disk makes it impr
  space required to store a table (in MB) is nodes * slices/node * columns. For example, on a 16 node cluster an er
  will occupy 640MB on disk.
- ⏱️ Query performance degrades significantly during data ingestion. WLM (Workload Management) tweaks help
  if you need consistent read performance, consider having replica clusters (at the extra cost) and swap them duri
- ❗ Never resize a live cluster. The resize operation can take hours depending on the dataset size. In rare cases, t
  stuck and you'll end up having a non-functional cluster. The safer approach is to create a new cluster from a sna
  cluster and shut down the old one.
- 🔸 Redshift has **reserved keywords** that are not present in Postgres (see full list here). Watch out for DELTA (Delt
- 🔸 Redshift does not support many Postgres functions, most notably several date/time-related and aggregation
  here.
- 🔸 Uniqueness, primary key, and foreign key constraints on Redshift tables are informational only and are not e
  used by the query optimizer to generate query plans. `NOT NULL` column constraints are enforced. See here for m
  defining constraints.
- 🔸 Compression on sort key can result in significant performance impact. So if your Redshift queries involving so
  might want to consider removing compression on a sort key.

- 🔹 Choosing a sort key is very important since you can not change a table's sort key after it is created. If you ne distribution key of a table, you need to create a new table with the new key and move your data into it with a qu new_table select * from old_table".
- ❗🚪 When moving data with a query that looks like "insert into x select from y", you need to have twice as mu table "y" takes up on the cluster's disks. Redshift first copies the data to disk and then to the new table. Here is a this for big tables.

## EMR

### EMR Basics

- 📒 Homepage · Release guide · FAQ · Pricing
- **EMR** (which used to stand for Elastic Map Reduce, but not anymore, since it now extends beyond map-reduce) i managed deployment of Hadoop, HBase and Spark. It reduces the management burden of setting up and maint yourself.

Back to top ⬆️

### EMR Alternatives and Lock-in

- ⛓️ Most of EMR is based on open source technology that you can in principle deploy yourself. However, the job tooling is AWS-specific. Migrating from EMR to your own clusters is possible but not always trivial.

Back to top ⬆️

### EMR Tips

- EMR relies on many versions of Hadoop and other supporting software. Be sure to check which versions are in u
- ⏱️ Off-the-shelf EMR and Hadoop can have significant overhead when compared with efficient processing on a data is small and performance matters, you may wish to consider alternatives, as this post illustrates.
- Python programmers may want to take a look at Yelp's mrjob.

- It takes time to tune performance of EMR jobs, which is why third-party services such as Qubole's data service a ways to improve performance or reduce costs.

Back to top ⬆️

### EMR Gotchas and Limitations

- 💸 ❗ **EMR costs** can pile up quickly since it involves lots of instances, efficiency can be poor depending on clust choice of workload, and accidents like hung jobs are costly. See the section on EC2 cost management, especially instances. This blog post has additional tips, but was written prior to the shift to per-second billing.
- 💸 Beware of "double-dipping". With EMR, you pay for the EC2 capacity and the service fees. In addition, EMR s means you pay for the storage and **PUT requests** at S3 standard rates. While the log files tend to be relatively sm depending on the size, generates thousands of log files that can quickly add up to thousands of dollars on the A aggregation is not available on EMR.

## Kinesis Streams

### Kinesis Streams Basics

- 📒 Homepage · Developer guide · FAQ · Pricing
- **Kinesis Streams** (which used to be only called Kinesis, before Kinesis Firehose and Kinesis Analytics were launche you to ingest high-throughput data streams for immediate or delayed processing by other AWS services.
- Kinesis Streams' subcomponents are called **shards**. Each shard provides 1MB/s of write capacity and 2MB/s of re of 5 reads per second. A stream can have its shards programmatically increased or decreased based on a variety
- All records entered into a Kinesis Stream are assigned a unique sequence number as they are captured. The reco ordered by this number, so any time-ordering is preserved.
- This page summarizes key terms and concepts for Kinesis Streams.

Back to top ⬆️

### Kinesis Streams Alternatives and Lock-in

- 🚪 Kinesis is most closely compared to Apache Kafka, an open-source data ingestion solution. It is possible to s hosted on EC2 instances (or any other VPS), however you are responsible for managing and maintaining both Zo brokers in a highly available configuration. Confluent has a good blog post with their recommendations on how links on the bottom to several other blogs they have written on the subject.
- ⛓️ Kinesis uses very AWS-specific APIs, so you should be aware of the potential future costs of migrating away f to use it.
- An application that efficiently uses Kinesis Streams will scale the number of shards up and down based on the re (Note there is no direct equivalent to this with Apache Kafka.)

Back to top ⬆️

## Kinesis Streams Tips

- The KCL (Kinesis Client Library) provides a skeleton interface for Java, Node, Python, Ruby and .NET programs to a Kinesis Stream. In order to start consuming data from a Stream, you only need to provide a config file to point Stream, and functions for initialising the consumer, processing the records, and shutting down the consumer wit
  - The KCL uses a DynamoDB table to keep track of which records have been processed by the KCL. This ensur processed "at least once". It is up to the developer to ensure that the program can handle doubly-processed
  - The KCL also uses DynamoDB to keep track of other KCL "workers". It automatically shares the available Kine workers as equally as possible.

Back to top ⬆️

## Kinesis Streams Gotchas and Limitations

- 🔸 ⏱️ Kinesis Streams' shards each only permit 5 reads per second. If you are evenly distributing data across ma for the Stream will remain at 5 reads per second on aggregate, as each consuming application will need to chec new records. This puts a hard limit on the number of different consuming applications possible per Stream for a latency.
  - For example, if you have 5 consuming applications reading data from one Stream with any number of shard latency of less than one second, as each of the 5 consumers will need to poll *each shard* every second, reach second per shard.

    - This blog post further discusses the performance and limitations of Kinesis in production.
- 💸 **Kinesis Streams are not included in the free tier.** Make sure if you do any experimentation with it on a perso down the stream or it may run up unexpected costs (~$11 per shard-month.)

## Kinesis Firehose

Back to top ⬆️

### Kinesis Firehose Gotchas and Limitations

- 🔸 📜 When delivering from Firehose to Elasticsearch, the JSON document cannot contain an "_id" property. Fi deliver those documents and won't log any error.

## Device Farm

### Device Farm Basics

- 📒 Homepage · Developer guide · FAQ · Pricing
- **Device Farm** is an AWS service that enables mobile app testing on real devices.
- Supports iOS and Android (including Kindle Fire) devices, as well as the mobile web.
- Supports remote device access in order to allow for interactive testing/debugging.

Back to top ⬆️

### Device Farm Tips

- AWS Mobile blog contains several examples of Device Farm usage for testing.
- Device Farm offers a free trial for users who want to evaluate their service.
- Device Farm offers two pricing models: Paying **per device minute** is useful for small usage levels or for situation usage amount. **Unmetered plans** are useful in situations where active usage is expected from the beginning.

- To minimize waiting time for device availability, one approach is to create several device pools with different dev
  choose one of the unused device pools on every run.

Back to top ⬆️

### Device Farm Gotchas and Limitations

- ❗ Devices don't have a SIM card and therefore can't be used for testing SIM card-related features.
- 🔸 Device Farm supports testing for most popular languages/frameworks, but not for all. An actual list of suppo
  languages is presented on this page.
- 🔸 The API and CLI for Device Farm is quite a low level and may require developing additional tools or scripts or
- 🔸 AWS provide several tools and plugins for Device Farm, however, it doesn't cover all cases or platforms. It ma
  specific tools or plugins to support specific requirements.
- ❗ In general, Device Farm doesn't have Android devices from Chinese companies like Huawei, Meizu, Lenovo, e
  supported devices located here.
- 🔸 Device availability is uneven. It depends on several factors including device popularity. Usually, more modern
  demand, thus the waiting time for them will be higher compared to relatively old devices.

## Mobile Hub

### Mobile Hub Basics

- 📒 Homepage · User guide · FAQ · Pricing

- **Mobile Hub** orchestrates multiple services to create an AWS backend for mobile and web applications.
- Each *project* in Mobile Hub has one *backend* made up of configurable features, plus one or more *applications*.
- Features include Analytics, Cloud Logic, Conversational Bots, Hosting and Streaming, NoSQL Database, User Dat
  Each feature uses one or two services to deliver a chunk of functionality.
- Services used include API Gateway, CloudFront, Cognito, Device Farm, DynamoDB, Lambda, Lex, Pinpoint and S3
- Application SDKs exist for Android (Java), iOS (Swift), Web (JS) and React Native (JS). There is also a CLI for JavaS

Back to top ⬆️

### Mobile Hub Tips

- The Mobile Hub console has starter kits and tutorials for various app platforms.
- The CLI allows local development of Lambda code (JS by default) with `awsmobile {pull|push}` commands, to sy
  and back again.
- Mobile Hub itself is free, but each of the services has its own pricing model.

Back to top ⬆️

### Mobile Hub Gotchas and Limitations

- 🔸 The Cloud API feature allows importing an existing Lambda function instead of defining a new one, but there
  with the CLI. Check the GitHub issues.
- ❗ Mobile Hub uses CloudFormation under the covers, and gets confused when a service is changed outside of

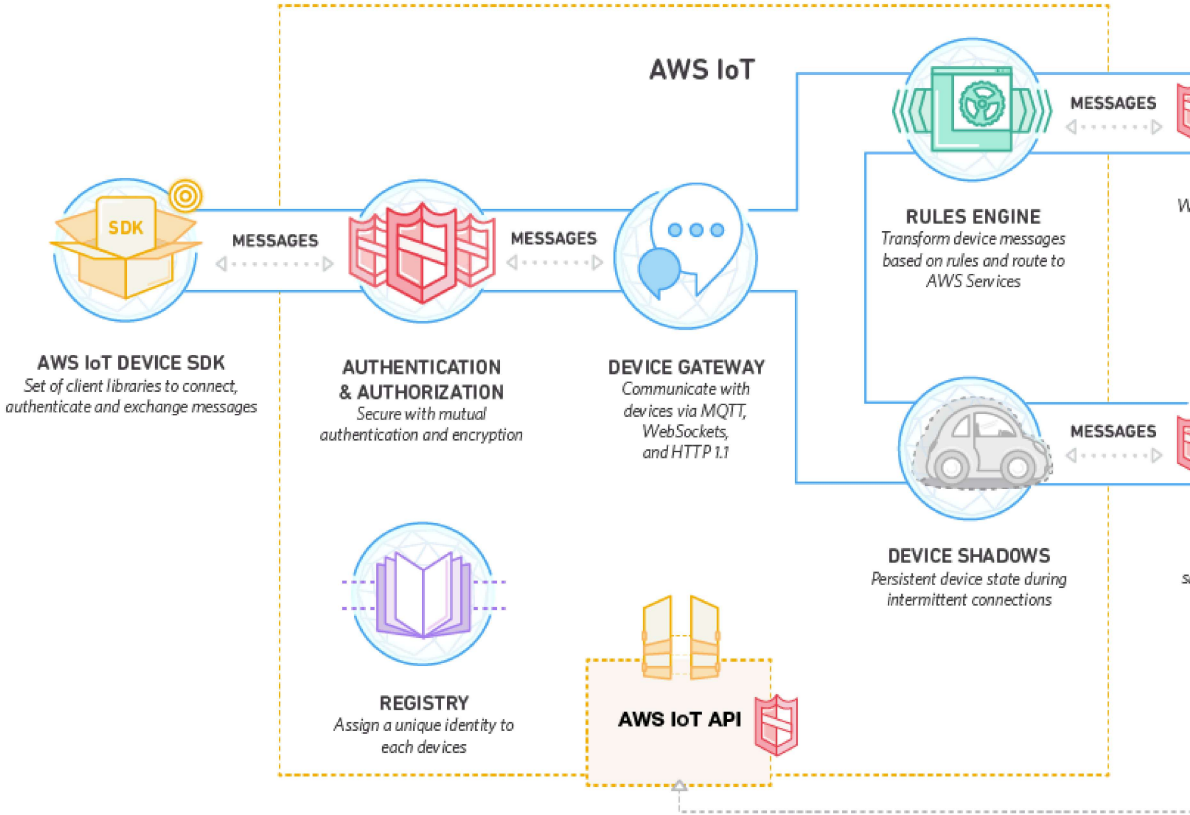
## IoT

### IoT Basics

- 📒 Homepage · User guide · FAQ · Pricing

- **IoT** is a platform for allowing clients such as IoT devices or software applications (examples) to communicate wit
- Clients are also called **devices** (or **things**) and include a wide variety of device types. Roughly there are three cat
  that interact with IoT services by sending message over an IoT protocol to the IoT Pub/Sub-style message broke
  **Device Gateway**:
  - Send messages only: For example, the AWS IoT Button on an eddystone beacon.
  - Send, receive, and process messages: For example, a simple processing board, such as a **Raspberry Pi** (quic
    device, such as the Echo or Echo Dot. These are designed to work with the Alexa skills kit, a programmable
- AWS has a useful quick-start (using the Console) and a slide presentation on core topics.

- **IoT terms:**
  - AWS **IoT Things** (metadata for devices in a registry) and can store device state in a JSON document, which i
    Device metadata can also be stored in **IoT Thing Types**. This aids in device metadata management by allow
    description and configuration for more than one device. Note that IoT Thing Types can be deprecated, but
    immutable.
  - AWS **IoT Certificates** (device authentication) are the logical association of a unique certificate to the logical
    This association can be done in the Console. In addition, the public key of the certificate must be copied to
    covers the authentication of devices to a particular AWS Device Gateway (or message broker). You can asso
    with an IoT device or you can register your own CA (Certificate Authority) with AWS, generate your own cer
    those certificates with your devices via the AWS Console or cli.
  - AWS **IoT Policies** (device/topic authorization) are JSON files that are associated to one or more AWS IoT cer
    associated devices to publish and/or subscribe to messages from one or more MQTT topics.
  - AWS **IoT Rules** are SQL-like queries which allows for reuse of some or all device message data, as described
    summarizes design patterns with for IoT Rules.
  - Shown below is a diagram which summarizes the flow of messages between the AWS IoT services:




Back to top ⬆️

## IoT Greengrass

- 📒 Homepage

- 🐥 **Greengrass** is a software platform that extends AWS IoT capabilities allowing Lambda functions to be run dir
  also enables IoT devices to be able to securely communicate on a local network without having to connect to th
  - Greengrass includes a local pub/sub message manager that can buffer messages if connectivity is lost so th
    messages to the cloud are preserved. Locally deployed Lambda functions can be triggered by local events,
    or other sources.
  - Greengrass includes secure authentication and authorization of devices within the local network and also be
    and the AWS cloud. It also provides secure, over-the-air software updates of Lambda functions.
- Greengrass core software includes a message manager object, Lambda runtime, local copy service for IoT Thing
  deployment agent to manage Greengrass group configuration.
- **Greengrass groups** are containers for selected IoT devices settings, subscriptions and associated Lambda functio
  a device is either a Greengrass core or an IoT device which will be connected that particular Greengrass core.
- The Greengrass core SDK enables Lambda functions to interact with the AWS Greengrass core on which they run
  messages, interact with the local Thing Shadows service, or invoke other deployed Lambda functions.
- The AWS Greengrass Core SDK only supports sending MQTT messages with QoS = 0.
- Shown below is a diagram which shows the architecture of AWS IoT Greengrass services:


Back to top ⬆️

## IoT Alternatives and Lock-in

- AWS, Microsoft and Google have all introduced IoT-specific sets of cloud services since late 2015. AWS was first,
  to general availability in Dec 2015. Microsoft released their set of IoT services for Azure in Feb 2016. Google has
  released their IoT services Android Things and Weave.
- Issues of lock-in center around your devices — protocols (for example MQTT, AMQP), message formats (such as
  security (certificates).

Back to top ⬆️

## IoT Tips

- **Getting started with Buttons:** One way to start is to use an **AWS IoT Button**. AWS provides a number of code sa
  IoT Button, you can use the AWS IoT console, click the "connect AWS IoT button" link and you'll be taken to the
  There you fill out your button's serial number to associate it with a Lambda. (As of this writing, AWS IoT buttons
  in the US.)
- **Connections and protocols:** It is important to understand the details of about the devices you wish to connect t
  including how you will secure the device connections, the device protocols, and more. Cloud vendors differ sign
  common IoT protocols, such as MQTT, AMQP, XMPP. AWS IoT supports **secure MQTT**, **WebSockets** and **HTTPS**.
- Support for **device security** via certificate processing is a key differentiator in this space. In August 2016, AWS ac
  registrations for IoT devices to their services.
- **Combining with other services:** It's common to use other AWS services, such as AWS Lambda, Kinesis and Dyna
  no means required. Sample IoT application reference architectures are in this screencast.
- **Testing tools:**
  - To get started, AWS includes a lightweight MQTT client in the AWS IoT console. Here you can create and tes
    messages to and from various MQTT topics.
  - When testing locally, if using MQTT, it may be helpful to download and use the open source Mosquitto bro
    with devices and/or device simulators
  - Use this MQTT load simulator to test device message load throughout your IoT solution.

Back to top ⬆️

## IoT Gotchas and Limitations

- 🔸 **IoT protocols:** It is important to verify the exact type of support for your particular IoT device message proto
  commonly used IoT protocol is MQTT. Within MQTT there are three possible levels of QoS in MQTT. AWS IoT su
  and forget, or at most once) and QoS 1(at least once, or includes confirmation), but *not* QoS 2 (exactly once, req
  This is important in understanding how much code you'll need to write for your particular application message r
  presentation about the nuances of connecting.
- 🔸 The ecosystems to match **IAM users or roles** to **IoT policies** and their associated authorized AWS IoT devices
  coding to enforce your security requirements is common.

- ❗ A common mistake is to misunderstand the importance of IoT **device security**. It is imperative to associate *ea* certificate (public key). You can generate your own certificates and upload them to AWS, or you can use AWS ge certificates. It's best to read and understand AWS's own guidance on this topic.
- 🔸 There is only one **AWS IoT Gateway** (endpoint) per AWS account. For production scenarios, you'll probably n accounts in order to separate device traffic for development, test and production. It's interesting to note that the supports configuration of multiple endpoints, so that a single Azure account can be used with separate pub/sub development, testing and production
- 🔸 **Limits:** Be aware of limits, including device message size, type, frequency, and number of AWS IoT rules.

Back to top ⬆️

### IoT Code Samples

- Simple Beer Service is a surprisingly useful code example using AWS IoT, Lambda, etc.
- IoT-elf offers clean Python sample using the AWS IoT SDK.
- IoT Button projects on Hackster include many different code samples for projects.
- 5 IoT code examples: a device simulator, MQTT sample, just in time registration, truck simulator, prediction data
- AWS Alexa trivia voice example is a quick-start using Alexa voice capability and Lambda.
- Some Raspberry Pi examples include the Beacon project, Danbo, and GoPiGo.

## SES

### SES Basics

- 📒 Homepage · Documentation · FAQ · Pricing
- **SES** (or Simple Email Service) is a service that exposes SMTP endpoints for your application to directly integrate

Back to top ⬆️

### SES Tips

- 🔹 **Bounce Handling:** Make sure you handle this early enough. Your ability to send emails can be removed if SES
- 🔹 **Credentials:** Many developers get confused between SES credentials and AWS API keys. Make sure to enter S using the SMTP APIs.

Back to top ⬆️

### SES Gotchas and Limitations

- 🔸 **Internet Access:** SES SMTP endpoints are on the Internet and will not be accessible from a location without In private subnet without NAT gateway route in the routing table). In such a case, set up an SMTP relay instance in access and configure your application to send emails to this SMTP relay instance rather than SES. The relay shou to send all emails to SES). ❗ If you are using a proxy instead of a NAT, confirm that your proxy service supports

## Certificate Manager

### Certificate Manager Basics

- 📒 Homepage · User guide · FAQ · Pricing
- Use the **Certificate Manager** to manage SSL/TLS certificates in other AWS services.
- Supports importing existing certificates as well as issuing new ones.
- Provides Domain Validated (DV) certificates. Validation can be done in two ways. The first (and recommended) w lives within Route 53 and the user has access, the necessary record can be added in the console via a single click request process. If the zone is not within Route 53 the user is required to update DNS manually. This is still prefe which requires more user interaction, and is done by sending an email to 3 contact addresses in WHOIS and 5 c domain, for each domain name present in the request.
- ACM will attempt to automatically renew a certificate issued by Amazon. It will first attempt to connect to the do that the certificate used by the domain is the same with the certificate that it intends to renew. Failing that, it wil used previously for validation. Failing that, ACM will attempt manual validation by sending emails to all domains

Back to top ⬆️

## Certificate Manager Alternatives and Lock-in

- ⛓️ Certificates issued by the Certificate Manager can't be used outside of the services that support it. Imported c still be used elsewhere.

Back to top ⬆️

## Certificate Manager Tips

- 🔹 **Supported services:** Managed Load Balancers, CloudFront, API Gateway and Elastic Beanstalk.
- 🔸 During the domain validation process, if DNS validation is unsuccessful Certificate Manager will send an ema specified in the domain's WHOIS record and up to five common administrative addresses. Some anti-spam filter because of this. You should check the spam folder of your email if you don't receive a confirmation email.
- 🔹 Setting up a certificate for a test domain you don't have email set up on? You can now use DNS validation in
- 🔹 Remember when requesting a wildcard domain that the request will not be valid for the level just below the w subdomains preceding the wildcard. Take for example an approved, issued certificate for `*.bar.example.com`. Th `foo.bar.example.com` but not `bar.example.com`. Likewise it would also not be valid for `www.bar.foo.example.com` each of these domains to the certificate request.

Back to top ⬆️

## Certificate Manager Gotchas and Limitations

- 🔸 In order to use **Certificate Manager** for CloudFront distributions, the certificate must be issued or imported f region.
- 🔸 Certificates used with Elastic Load Balancers must be issued in the same region as the load balancer. Certifica copied between regions, as of July 2017. If a domain uses load balancers present in multiple regions, a different requested for each region.
- 🔸 **IoT** has its own way of setting up certificates.
- 🔸 By default the maximum number of domains per certificate is 10. You can get this limit increased to a maximu AWS support. **Note** for every different domain you have on the requested cert, you'll need to press accept on an For example if you request a cert with 42 different domains or sub domains, you'll need to press accept on 42 di

    - 🔹 If you request a limit increase to AWS support for this, they will respond to you asking to confirm this. By body of your initial request: `"I acknowledge at the moment, there is no method to add or remove a name f Instead, you must request a new certificate with the revised namelist and you must then re-approve al certificate, even if they'd been previously approved."`
- 🔸 There is no way at the moment to add or remove a domain to/from an existing certificate. You must request approve it from each of the domains requested.

## WAF

### WAF Basics

- 📒 Homepage · Documentation · FAQ · Pricing
- WAF (Web Application Firewall) is used in conjunction with the CloudFront and ALB services to inspect and block on user-configurable conditions.
- HTTPS and HTTP requests are supported with this service.
- WAF's strength is in detecting malicious activity based on pattern-matching inputs for attacks such as SQL inject
- WAF supports inspection of requests received through both IPv6 and IPv4.

Back to top ⬆️

### WAF Tips

- Getting a WAF API call history can be done through CloudTrail. This is enabled through the CloudTrail console.
- It's also possible to get full logs of all the web requests inspected

Back to top ⬆️

### WAF Gotchas and Limitations

- As of May 2019, AWS WAF is available on Amazon CloudFront and in 12 commercial AWS regions: US East (N. V West (Oregon), US West (N. California), EU (Ireland), EU (Frankfurt), EU (London), EU (Stockholm), Asia Pacific (To

Asia Pacific (Singapore), and Asia Pacific (Seoul).

# OpsWorks

## OpsWorks Basics

- 📒 Homepage · Documentation · FAQ · Pricing: Stacks, Chef Automate, Puppet Enterprise
- OpsWorks is a configuration management service that uses Chef or Puppet configuration management. It is bro
  services:
  - OpsWorks Stacks: The service lets you configure and launch stacks specific to your application's needs, and
    application deployments. Chef runs can be performed manually via the Execute Cookbooks command, othe
    part of lifecycle events.
    - OpsWorks Stacks differs from standard configuration management services in that it also allows you to
      infrastructure and application automation (such as creating Amazon EC2 instances and deploying appli
      cookbooks).
  - OpsWorks for Chef Automate: This service launches a dedicated Chef Automate server in your account, whi
    nodes, upload cookbook code, and configure systems. Automated patching, backups, OS updates, and min
    are provided as part of the service. An AWS API is provided for associating/disassociating nodes. Chef runs
    using the chef-client cookbook.
  - OpsWorks for Puppet Enterprise: This service launches a dedicated Puppet Master in your account, which ca
    nodes, upload modules, and configure systems. Automated patching, backups, OS updates, and minor Pupp
    provided as part of the service. An AWS API is provided for associating/disassociating nodes. By default, the
    automatically every 30 minutes on associated nodes.
- OpsWorks for Chef Automate and OpsWorks for Puppet Enterprise are strictly designed for configuration manag
  provision infrastructure outside the Chef Server/Puppet Master that is created in our account.
- All three OpsWorks services support managing both Amazon EC2 and on-premises infrastructure, however the i
  differ slightly.
  - OpsWorks Stacks allows you to register instances and install the OpsWorks Agent to connect to your stack.
  - OpsWorks for Chef Automate and OpsWorks for Puppet Enterprise allow you to associate new or existing in
    the opsworks-cm:AssociateNode API action or the vendor-supported method for associating nodes to Chef

Enterprise.

- Although OpsWorks will let you work with common Chef recipes or Puppet modules when creating your stacks, will require familiarity with Chef or Puppet syntax. Chef/Puppet code is not supported as part of AWS Support.
- As of December 2016, OpsWorks Stacks supports Chef versions 12, 11.10.4, 11.4.4 and 0.9.15.5.
- As of December 2016, OpsWorks for Chef Automate uses Chef Server version 12.11.1 This is the current stable v
- Berkshelfcan be used with Chef stacks of version 11.10 and later for managing cookbooks and their respective d Chef 12.x stacks, Berkshelf must be installed by the stack administrator.
- Running your own Chef environment may be an alternative to consider - some considerations are listed in this E

Back to top ⬆️

## OpsWorks Alternatives and Lock-in

- Major competitors in Configuration Management include:
  - Chef
  - Puppet
  - Ansible.

Back to top ⬆️

## OpsWorks Tips

- OpsWorks Stacks and OpsWorks for Chef Automate use Chef cookbooks for configuration. Chef provides free tr practices, etc. at https://learn.chef.io.
- OpsWorks for Puppet Enterprise uses Puppet manifests for configuration. Puppet provides a very useful learning https://learn.puppet.com/.

Back to top ⬆️

## OpsWorks Gotchas and Limitations

- OpsWorks Stacks is not available in the following regions:

    - Montreal
    - GovCloud
    - Beijing
- OpsWorks for Chef Automate and OpsWorks for Puppet Enterprise are not available in the following regions:
    - Montreal
    - Sao Paulo
    - GovCloud
    - London
    - Paris
    - Seoul
    - Mumbai

# Batch

## Batch Basics

- 📒 Homepage · Documentation · FAQ · Pricing
- **AWS Batch** is a service that offers an environment to run batch computing jobs. The service dynamically provisio resources needed by the jobs based on their resource requirements, and can scale up to hundreds of thousands
- These batch workloads have access to all other AWS services and features.
- AWS Batch, coupled with spot instances can help run the jobs when appropriate capacity is available, providing compute resources.
- The batch workloads are built as a Docker Image. These images can then pushed to the EC2 Container Registry repository that can be accessed from AWS.
- A Job Definition has the workload's Docker Image URI, and also lets the users specify the environment details lik mappings, environment variables, parameters, retry strategy, container properties, and the job's IAM role.
- The Compute Environments are EC2 clusters that provide the runtime for the batch workloads to execute in.

- AWS Batch provides managed, as well as unmanaged compute environments. The Managed Environments are p by AWS, while the Unmanaged Environments are managed by the customers.
- The Job Definitions are submitted to Job Queue(s) for execution. Each queue has a priority, and has at least one associated with it.
- AWS Batch uses ECS to execute the containerized jobs.

Back to top ⬆️

### Batch Tips

- AWS Batch supports prioritization of jobs via the Job Queue Priority. Higher the number - higher the priority.
- AWS Batch supports launching the Compute Environment into specific VPC and subnets.
- A Compute Environment is same as an ECS Cluster.
- There is no additional cost for AWS Batch. You only pay the cost associated with the AWS Services being used - resources consumed by the batch jobs.
- Associate IAM Roles and policies with the Compute Environment to enable the containers access to other AWS r
- 🔹 Use Unmanaged Compute Environments if you need specialized resources like Dedicated Hosts, or EFS.

## SQS

### SQS Basics

- 📒 *Homepage* · *Documentation* · *FAQ* · *Pricing*
- SQS is a highly scalable, fully managed message queuing service from AWS.
- SQS supports the pull model, where the producers *queue* the messages, and the consumers pull messages off th
- SQS provides a message visibility timeout, during which the message being processed will not be delivered to o consumer does not delete the message after processing, the message becomes available to other consumers up visibility timeout. This parameter is called VisibilityTimeout.
- Each message can have up to 10 custom fields, or attributes.

- SQS allows producers to set up to 15 minutes of delay before the messages are delivered to the consumers. This DelaySeconds.
- There are two types of queues supported by SQS -
  - Standard Queues
    - Guarantee **at least once** delivery of the messages.
    - Do not retain the order of delivery of the messages.
  - FIFO Queues
    - Guarantee **only once** delivery of the messages
    - Guarantee the order of the delivery of the messages
- SQS supports fine grained access to various API calls and Queues via IAM policies.
- The messages that fail to process can be put in a dead letter queue.

Back to top ⬆️

## SQS Alternatives and Lock-In

- Alternatives to SQS include Kafka, RabbitMQ, ActiveMQ and others.
- Google Cloud Platform has Pub/Sub, and Azure has Azure Queue Service.
- SQS vs SNS

Back to top ⬆️

## SQS Tips

- SNS can be used in combination of SQS to build a "fan out" mechanism by having an SQS Queue subscribe to th
- SQS supports encryption using AWS KMS.
- Cloudwatch alarms can be creating using various SQS metrics to trigger autoscaling actions and/or notifications

Back to top ⬆️

## SQS Gotchas and Limitations

- 🔸 SQS does not have a VPC endpoint (unlike S3 and DynamoDB), so SQS will need to be accessed using public
- 🔸 FIFO Queues are limited to 300 API calls per second.
- 🔸 FIFO Queues cannot subscribe to an SNS topic.
- 🔸 Standard Queues can deliver duplicate messages regardless of the visibility window. If only-once delivery is y
  FIFO queues, or build an additional layer to de-dupe the messages.
- 🔸 You can send/receive messages in batch, however, there can only be maximum of 10 messages in a batch.

## SNS

### SNS Basics

- 📒 *Homepage* · *Documentation* · *FAQ* · *Pricing*
- **SNS** (Simple Notification Service) is a pub/sub based, highly scalable, and fully managed messaging service that
  mobile notifications.
- SNS can push the messages down to the subscribers via SMS, Email, SQS, and HTTP/S transport protocols.
- Producers publish messages to a SNS Topics, which can have many subscribers.
- Each subscription has an associated protocol, which is used to notify the subscriber.
- A copy of the message is sent to each subscriber using the associated protocol.
- SNS can also invoke lambda functions.

Back to top ⬆️

### SNS Alternatives and Lock-In

- Popular alternatives to SNS are Kafka, Notification Hubs on Azure, and Pub/Sub on Google Cloud.
- **SNS vs SQS:**
  - Both SNS and SQS are highly scalable, fully managed messaging services provided by AWS.
  - SQS supports a *pull* model, while SNS supports a *push* model. Consumers have to pull messages from an SQ
    pushed the message from an SNS Topic.

- An SQS message is intended to be processed by only one subscriber, while SNS topics can have many subsc
- After processing, the SQS message is deleted from the queue by the subscriber to avoid being re-processed
- An SNS message is *pushed* to all subscribers of the topic at the same time, and is not available for deletion
- SNS supports multiple transport protocols of delivery of the messages to the subscribers, while SQS subscri messages off the queue over HTTPS.

Back to top ⬆️

### SNS Tips

- Fan-out architecture can be achieved by having multiple subscribers for a topic. This is particularly useful when e out to multiple, isolated systems.
- SNS topics can be used to power webhooks with backoff support to subscribers over HTTP/S.
- SQS queues can subscribe to SNS topics.
- SNS is used to manage notifications for other AWS services like Autoscaling Groups' notifications, CloudWatch A
- SNS is frequently used as "glue" between disparate systems— such as GitHub and AWS services.

Back to top ⬆️

### SNS Gotchas and Limitations

- 🔸 HTTP/S subscribers of SNS topics need to have public endpoints, as SNS does not support calling private en private subnet within a VPC).
- 📜 In a fan-out scenario, SSE-enabled SQS subscribers of an SNS topic will not receive the messages sent to the

## High Availability

This section covers tips and information on achieving high availability.

### High Availability Tips

- AWS offers two levels of redundancy, regions and availability zones (AZs).
- When used correctly, regions and zones do allow for high availability. You may want to use non-AWS providers f mitigation (i.e. not tying your company to one vendor), but reliability of AWS across regions is very high.
- **Multiple regions:** Using multiple regions is complex, since it's essentially like managing completely separate infr for business-critical services with the highest levels of redundancy. However, for many applications (like your ave deploying extensive redundancy across regions may be overkill.
- The High Scalability Blog has a good guide to help you understand when you need to scale an application to mu
- 🔹 **Multiple AZs:** Using AZs wisely is the primary tool for high availability!
  - A typical single-region high availability architecture would be to deploy in two or more availability zones, w as in this AWS diagram.
  - The bulk of outages in AWS services affect one zone only. There have been rare outages affecting multiple z example, the great EBS failure of 2011) but in general most customers' outages are due to using only a sing infrastructure.
  - Consequently, design your architecture to minimize the impact of AZ outages, especially single-zone outag
  - Deploy key infrastructure across at least two or three AZs. Replicating a single resource across more than th make sense if you have other backup mechanisms in place, like S3 snapshots.
  - A second or third AZ should significantly improve availability, but additional reliability of 4 or more AZs may complexity (unless you have other reasons like capacity or Spot market prices).
  - 💸 Watch out for **cross-AZ traffic costs**. This can be an unpleasant surprise in architectures with large volum boundaries.
  - Deploy instances evenly across all available AZs, so that only a minimal fraction of your capacity is lost in ca
  - If your architecture has single points of failure, put all of them into a single AZ. This may seem counter-intu likelihood of any one SPOF to go down on an outage of a single AZ.
- **EBS vs instance storage:** For a number of years, EBSs had a poorer track record for availability than instance stor individual instances can be killed and restarted easily, instance storage with sufficient redundancy could give hig has improved, and modern instance types (since 2015) are now EBS-only, so this approach, while helpful at one archaic.
- Be sure to use and understand CLBs/ALBs appropriately. Many outages are due to not using load balancers, or n misconfiguring them.

Back to top ⬆️

### High Availability Gotchas and Limitations

- 🔸 **AZ naming** differs from one customer account to the next. Your "us-west-1a" is not the same as another cust letters are assigned to physical AZs randomly per account. This can also be a gotcha if you have multiple AWS a IDs are consistent between accounts, and can be used to reliably align between AWS accounts.
- 🔸 💸 **Cross-AZ traffic** is not free. At large scale, the costs add up to a significant amount of money. If possible, within the same AZ as much as possible.

## Billing and Cost Management

### Billing and Cost Visibility

- AWS offers a **free tier** of service, that allows very limited usage of resources at no cost. For example, a micro inst storage is available for no charge. Many services are only eligible for the free tier for the first twelve months that other services offer a free usage tier indefinitely. (If you have an old account but starting fresh, sign up for a new tier.) AWS Activate extends this to tens of thousands of dollars of free credits to startups in certain funds or acce
- You can set **billing alerts** to be notified of unexpected costs, such as costs exceeding the free tier. You can set th
- AWS offers Cost Explorer, a tool to get better visibility into costs.
- Unfortunately, the AWS console and billing tools are rarely enough to give good visibility into costs. For large ac console can time out or be too slow to use.
- **Tools:**
  - 🔹 Enable billing reports and install an open source tool to help manage or monitor AWS resource utilizatio written by Netflix) is probably the first one you should try. Check out docker-ice for a Dockerized version th
  - 🔸 One challenge with Ice is that it doesn't cover amortized cost of reserved instances.
  - Other tools include Security Monkey and Cloud Custodian.
  - Use AWS Simple Monthly Calculator to get an estimate of usage charges for AWS services based on certain Monthly charges will be based on your actual usage of AWS services, and may vary from the estimates the

- **Third-party services:** Several companies offer services designed to help you gain insights into expenses or lowe Cloudability, CloudHealth Technologies, and ParkMyCloud. Some of these charge a percentage of your bill, whic the market landscape.
- AWS's Trusted Advisor is another service that can help with cost concerns.
- Don't be shy about asking your account manager for guidance in reducing your bill. It's their job to keep you ha
- **Tagging for cost visibility:** As the infrastructure grows, a key part of managing costs is understanding where the to tag resources, and as complexity grows, group them effectively. If you set up billing allocation appropriately, into expenses according to organization, product, individual engineer, or any other way that is helpful.
- If you need to do custom analysis of raw billing data or want to feed it to a third party cost analysis service, enab report feature.
- Multiple Amazon accounts can be linked for billing purposes using the Consolidated Billing feature. Large enterp billing structures depending on ownership and approval processes.
- Multiple Amazon accounts can be managed centrally using AWS Organizations.

Back to top ⬆️

## AWS Data Transfer Costs

- For deployments that involve significant network traffic, a large fraction of AWS expenses are around data transf data transfer, within AZs, within regions, between regions, and into and out of AWS and the internet vary signific deployment choices.
- Some of the most common gotchas:
  - 🔸 *AZ-to-AZ traffic:* Note EC2 traffic between AZs is effectively the same as between regions. For example, c cluster across AZs is helpful for high availability, but can hurt on network costs.
  - 🔸 *Using public IPs when not necessary:* If you use an Elastic IP or public IP address of an EC2 instance, you w even if it is accessed locally within the AZ.
  - 🔸 *Managed NAT Gateway data processing:* Managed NAT Gateways are used to let traffic egress from priva 4.5¢ as a data processing fee layered on top of data transfer pricing. Past a certain point, running your own more cost effective.

- *Some services do cross-AZ traffic for free:* Many AWS services you'd not consider on their own merits offe
  cross-AZ data transfer. EFS, RDS, MSK, and others are examples of this.

- This figure gives an overview: