EBOOK

How to diagram your cloud architecture: Best practices from AWS solutions architects

James Wenzel Principal Solutions Architect, AWS

Jason Mimick Senior Partner Solutions Architect, AWS

Colten Woo Product Marketing Manager, Datadog

datadog.com

About the authors

Jamie Wenzel
Principal Solutions Architect
AWS

Jamie Wenzel is a Principal Solutions Architect networking specialist in the AWS EC2 Networking team. Jamie is part of the application networking organization and contributes to the design of application networking products and services. He is an avid public speaker and has presented at various AWS venues, including re:Invent, re:Inforce, lofts, and summits, as well as through AWS's Twitch channel. He has been with Amazon for more than six years and is passionate about helping people and organizations in their cloud journeys.

Jason Mimick
Senior Partner Solutions Architect
AWS

Jason Mimick is a Senior Partner Solutions Architect at AWS, specializing in cloud native technologies, DevOps, and observability. Jason has been working in commercial software since the late nineties, spending time as an engineer, architect, and product manager for select companies such as Microsoft and MongoDB before joining AWS.

Colten Woo
Product Marketing Manager
Datadog

Colten Woo is a Product Marketing Manager at Datadog. He owns go-to-market strategies for Cloudcraft and Datadog Metrics. Colten works with Cloudcraft customers to understand their use cases and best practices for diagramming cloud architecture.

Preface

Organizations are increasingly looking to migrate to the cloud or build cloud-native applications that will scale with their business. To accomplish these objectives, organizations often require many different views into the layers of their existing environments in order to get a clear understanding of where they are and how they're performing.

An infrastructure-level view can provide high-level data points such as latency and cost. But a secondary view is needed at the traffic level as well. This can visualize the actual movement of information across services, accounts, and regions. If you can't see how traffic moves, you can't truly calculate cost or detect latency issues. And if you just focused on telemetry, you wouldn't be able to look across 10s or 100s of services, infrastructure assets, and several accounts. For example, an underused or orphan service can lead to an unnecessary security exposure or unnecessary costs. In another case, a dataset that is not meant to be accessed or manipulated due to compliance protocols could be stored locally or accessed by a developer from another team who is not aware of these protocols. Miscommunications like these are not only an inconvenience to development teams but can lead to critical issues that impact end-users and customers.

Having a tool to clearly visualize your environment(s) is critical because it allows you to detect anomalies, create a shared view and alignment across teams, and understand your spend as you build new applications in the cloud.

Key cloud-architecting principles to keep in mind are:

- Security
- Scalability
- Availability
- Visibility

Different stakeholders may prioritize these objectives and principles differently. For example, solutions architects are focused on having clear visibility into the architecture. They are usually evangelizing new solutions, so the more simple they can make their architecture, the better. Developers and DevOps teams are focused on shipping applications that are secure and performant. They're also looking to get a holistic view of applications, understand how applications communicate with each other, and to drill down into details. Unlike solution architects, DevOps teams and SREs are focused on troubleshooting issues which are often caused by code bugs or outages. Lastly, leaders and CTOs are looking to understand their resources, drive operational excellence, and control overall spend.

This e-book will provide top-level practices for best of breed observability across all your computing environments. To lay out these practices, let's first align on top goals of the personas we highlighted above. These are high level developer objectives that are critical to a successful cloud adoption and can help us keep in mind what is important as we discuss our best practices.

- Reducing time to market or launch software
- Reducing IT spend
- Increasing IT uptime
- Strengthening security posture
- Increasing cross-team communication

With these goals in mind, let's dive into our top six best practices for diagramming your cloud architectures.

1

Start wide and dive deeper into resources

A cloud diagram can be useful for many use cases: cloud migrations, audits, troubleshooting, cost optimization, or architecture ideation. In all of these cases, it is important to first inventory everything in your environment. Whether you're a developer or solutions architect, the first question you should answer is: what is already running in my cloud environment? Having an overview of all the resources and relationships in your environment can help you spot any unused, underutilized, or misconfigured resources. You can take it a step further by visualizing multi-region connections and relationships to ensure optimal performance and resilience, as well as reduce costs, by bringing together resources from different zones into a single view.

Once you have a high-level overview of your architecture, you can filter for specific services, tags, resource-types, and accounts. With these filters in place, you can customize views to contain only relevant resources to spot anomalous ones that don't belong such as legacy or "temporary" infrastructure and identify security misconfigurations or vulnerable attack paths. Lastly, visualizing how network traffic moves through your environment is critical for understanding traffic patterns and maintaining a reliable environment. This can highlight bottlenecks and single points of failure, and identify threat surfaces to your workload.

2

Enforcing security for your diagrams is important to avoid revealing sensitive info

When looking for a cloud-architecting tool, the highest priority for developers, solutions architects, and site reliability engineers should be security and compliance. All data, including your diagrams, should stay private by default and only be accessible by you or other authorized users. The processes for explicitly sharing diagrams, revoking access, and creating role-based access controls for teams must also be seamless.

Additionally, you should look for tools that can be used by security teams to surface threats and security misconfigurations to minimize vulnerabilities. Security teams can leverage diagrams to meet compliance standard requirements such as SOC2 by documenting their architecture to visualize security groups and understand which services are connected through inbound and outbound rules. Maintaining an archive of each version of a diagram also ensures teams are prepared for audits and enables them to compare configurations to identify any changes that may have mistakenly introduced security exposures.

3

Plan with cloud budget in mind

When it comes to cloud architecting, cost considerations are often left to the very end. This is a mistake that can cause you to go back to the drawing board in the future. Having a diagramming tool that surfaces cost data in real-time for new projects is an invaluable tool for companies to make fully informed decisions about their architecture. When planning out architecture solutions, being able to see cost and performance metrics in a single place is also useful for understanding potential tradeoffs between cost, scalability, and performance. Additionally, having this visibility as you iterate through various permutations of the architecture can help you choose the correct solution for your cost and performance needs before you deploy.

Developers and solutions architects should work together to understand their current cloud costs, their budget, and how their new builds will impact their future spend.

4

Get feedback and iterate with your team

You might have a great system architecture idea in your head, but it can be very difficult to get it on paper in an easy-to-understand format. What helps most is creating a shared view with your peers, whether they're in the engineering or architecture organization. While sharing, it's also important to maintain control over your diagrams by sharing them with read-only access to ensure reviewers don't inadvertently make changes.

Adhering to consistent design frameworks and guidelines is also highly recommended so there's a common understanding of how to represent the various components of your architecture. This is especially true when onboarding new team members, so they can quickly get started without having to define–or redefine–what each component represents.

Having multiple versions of your diagrams can also improve clarity by providing a tailored view of your architecture based on the resources different teams are responsible for and care about. These should not be circulated outside of the teams that need them. You can always have other diagrams for presentations, leadership, etc. with personalized degrees of complexity or information as needed by the audience.

5

Keep your cloud diagrams up-to-date as a source of truth

A common challenge across developers and solutions architect teams is the issue of outdated diagrams. Static diagrams require hours of manual maintenance to avoid becoming outdated. Troubleshooting also becomes time-consuming when using manual reports that may have outdated information. To address this, live diagrams can be used to save time on manual development and ensure teams are always working off the latest version of your architecture. These live diagrams can then be embedded in internal documentation pages (e.g., Confluence) so they're readily accessible, improving cross-team alignment with a single source of truth.

This way, teams have access to diagrams that dynamically update as you make changes in your cloud environment. You can also create a complete archive of your diagrams by using version history to track revisions and revert to past versions.

6

Present professional diagrams backed by data to build stakeholder trust

Having diagrams that stand out and are not "ordinary" is important. This is especially true if you are a solutions architect; you want your clients and development stakeholders to be captivated. Visually impressing your audience is one major step in building trust and interest in your ideas. You also want to be confident and pleased to be putting your diagrams in front of your audience.

Professional diagrams can establish your credibility and allow you to go from idea to presentation in minutes. Data points are also critical for leadership support. You should have key metrics and logs ready to show security detection or change in cost over time.

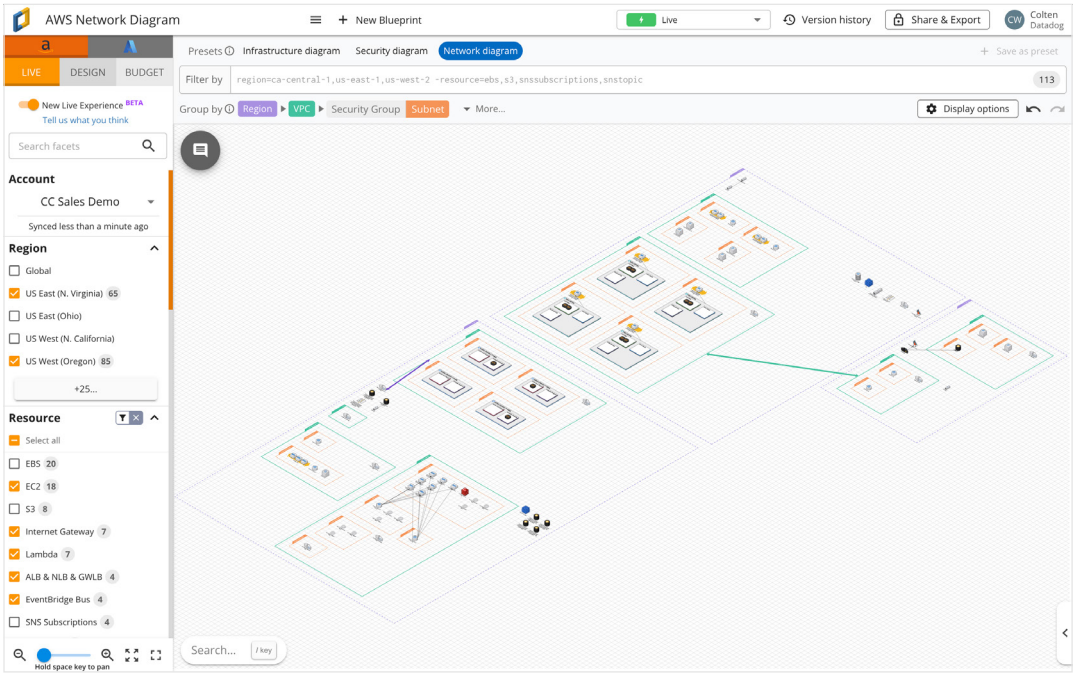
About Cloudcraft

Before you can implement these guidelines, you need to select the right solution for your needs. Whether you're building a cloud architecture from scratch, tracking changes in your environment, or looking to optimize cloud cost, Cloudcraft allows you to visually understand and communicate your architecture with ease. We'll now cover how Cloudcraft supports the best practices outlined above.

Live

The Live tab in Cloudcraft automatically generates live diagrams of your architecture to help you produce real-time documentation, prepare for audits, troubleshoot your environment, and inventory all your resources.

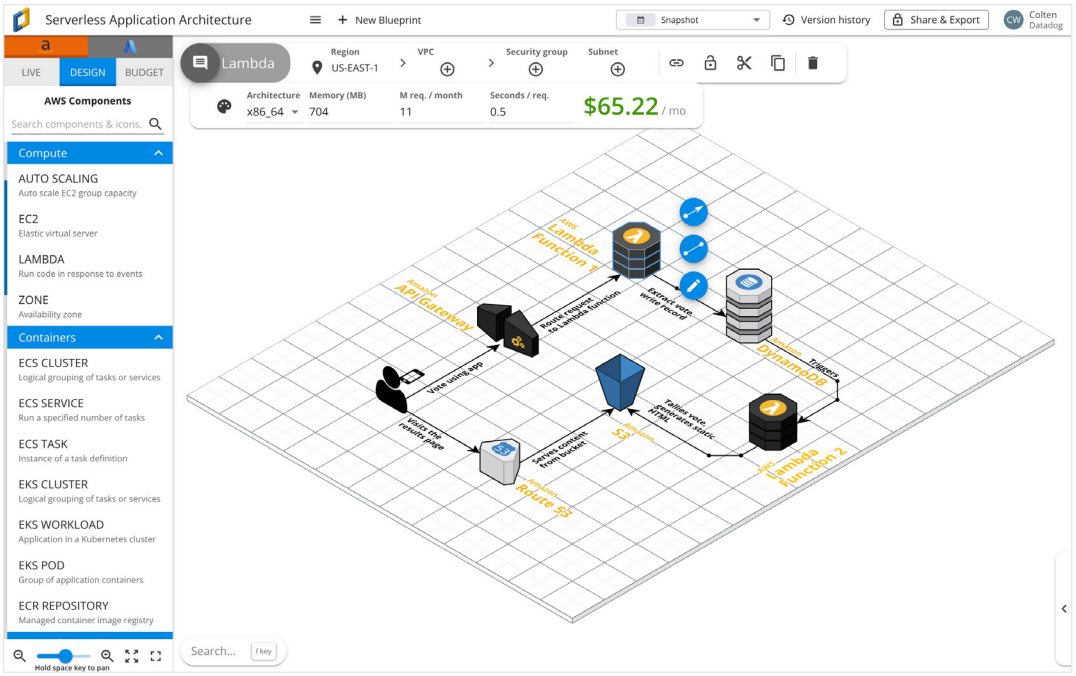
- Securely connect to your AWS and Azure accounts through read-only roles
- Visualize cloud resources and how they're connected with preset views for infrastructure, security, and network diagrams
- Populate an inventory list of all the resources and tags in your environment then use them as filters to fine-tune your diagram
- Access detailed information about a resource including its attributes, configuration, and live metrics with a single click
- Seamlessly pivot to related views in Datadog to gain additional context and advance investigations



Design

The Design tab in Cloudcraft helps you rapidly design and edit architectures in a collaborative environment to help you plan new architectures, create project proposals, and prepare for cloud migrations.

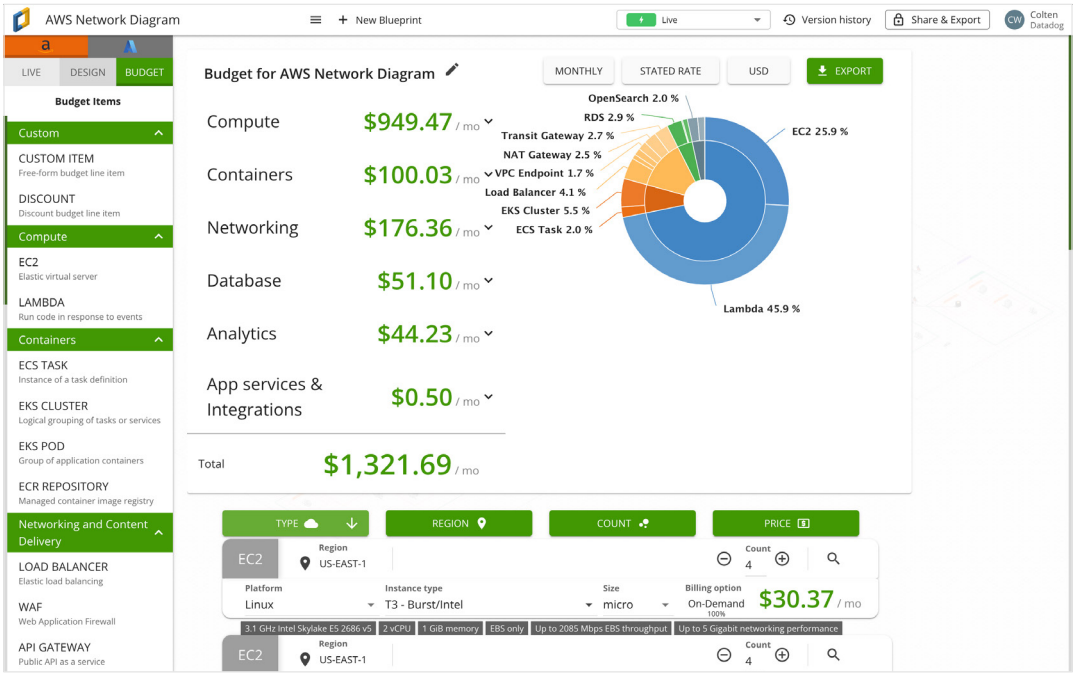
- Access hundreds of preset AWS and Azure components or add custom icons for additional resources
- Set the attributes for a component and get dynamic cost estimates based on its configuration
- Organize components by grouping AWS resources by region, VPC, security group, and subnet and Azure Resources by region, resource group, VNet, NSG, and subnet
- Visualize relationships between resources by drawing arrow connections



Budget

The Budget tab provides cost estimates using the AWS Cost API and Azure Prices API to help you forecast, report, and optimize your cloud spend.

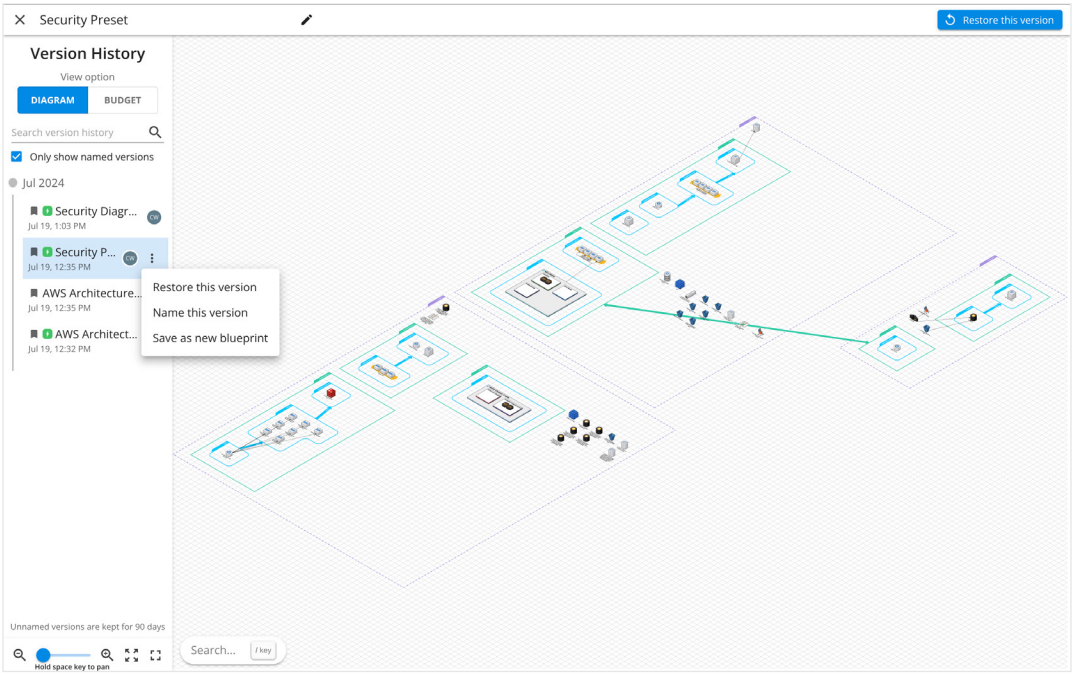
- Understand how spend is allocated across categories and resources to identify costly areas
- Hone in on resource-level costs and adjust attributes to explore cost saving opportunities in a read-only format
- Add custom line items and discounts to properly reflect the terms of your contract for an accurate cost estimate



Platform

All three areas of Cloudcraft are built on top of a secure and flexible platform built for enterprise scale.

- Ensure a seamless and secure authentication experience through single sign-on and multi-factor authentication
- Create and update diagrams programmatically using the Cloudcraft API
- Track changes, name important milestone versions, and revert to previous iterations with version history
- Collaborate across teams seamlessly through simple team management, comments, and multi-user editing
- Export diagrams or share a live view by providing a direct link or embedding it in your documentation tool of choice



Start your free trial today

DATADOG