Amazom Q with MCP Servers

What is LLM?

-> A Large Language Model (LLM) is a type of AI that uses natural language processing (NLP) to gain insights from text. It can generate text, summarize content and helps for QA search using NLP.

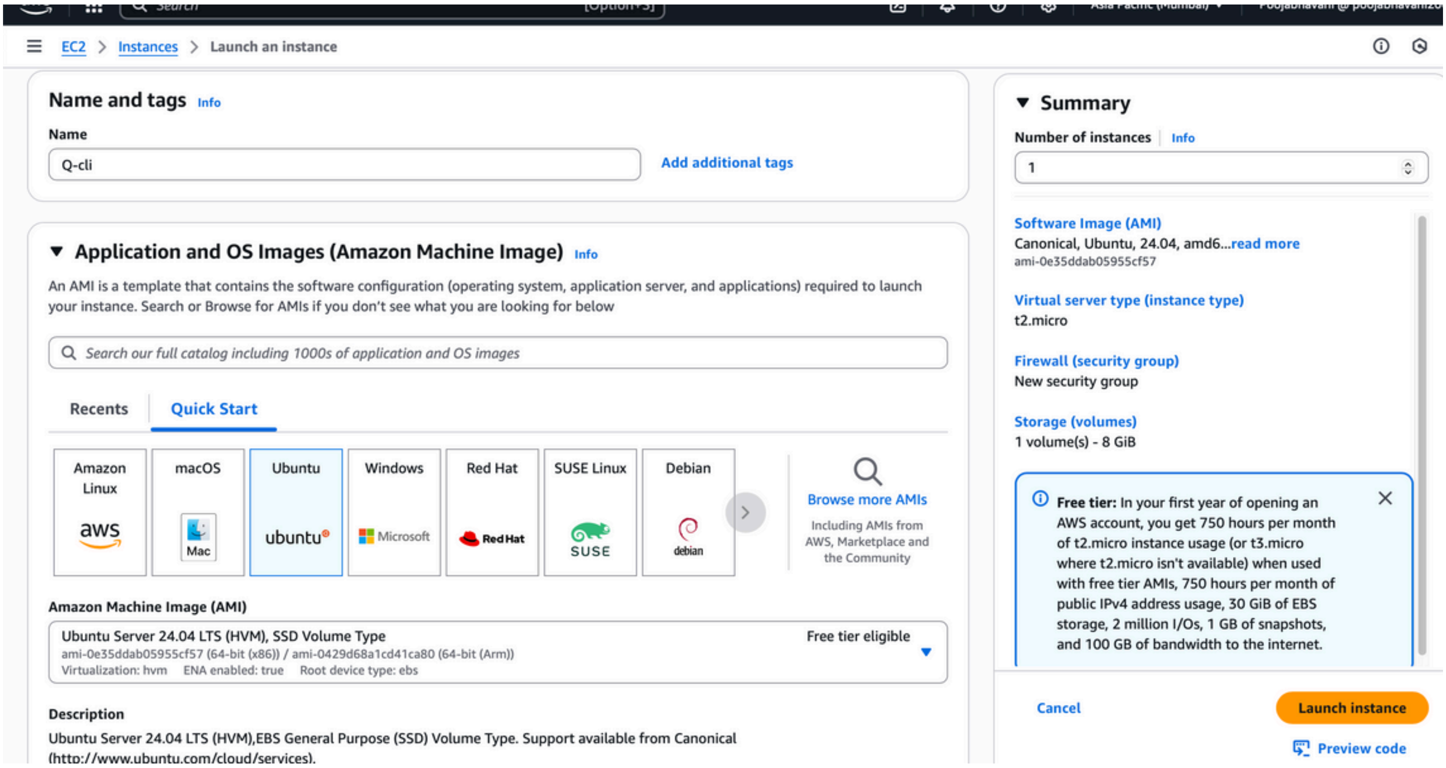
What is Amazon Q Developer?

-> Is generative AI-powered assistant designed to increase productivity. It can assist with tasks such as code generation, debugging, documentation, and answering technical questions, making development faster and more efficient.

Using Amazon Q with MCP Servers

For Commands

https://community.aws/content/2x5FTH4rEuKtWz1iRm8RwT9wJq5/installing-amazon-q-cli-setting-up-mcp-server



Name and tags Info

Name: Q-cli

Add additional tags

Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type — Free tier eligible
ami-0e35ddab05955cf57 (64-bit (x86)) / ami-0429d68a1cd41ca80 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Ubuntu Server 24.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).

Summary

Number of instances Info: 1

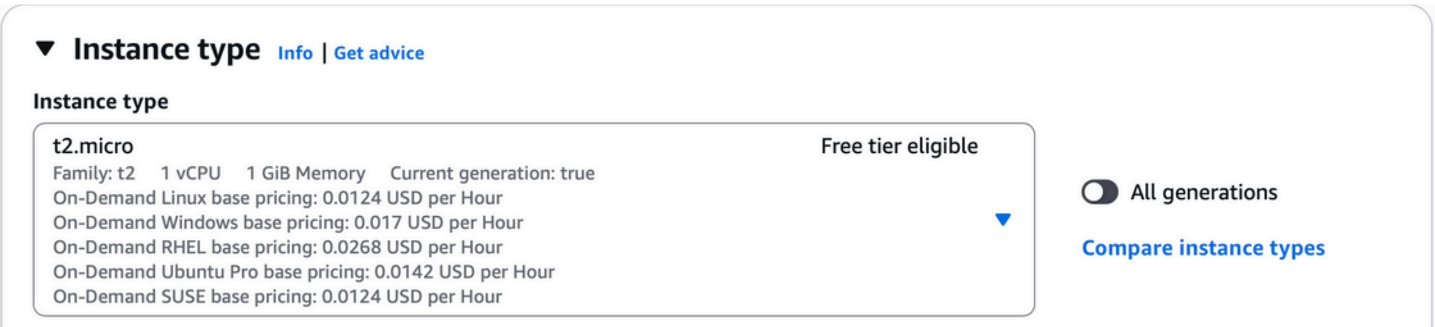
Software Image (AMI): Canonical, Ubuntu, 24.04, amd6...read more
ami-0e35ddab05955cf57

Virtual server type (instance type): t2.micro

Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 8 GiB

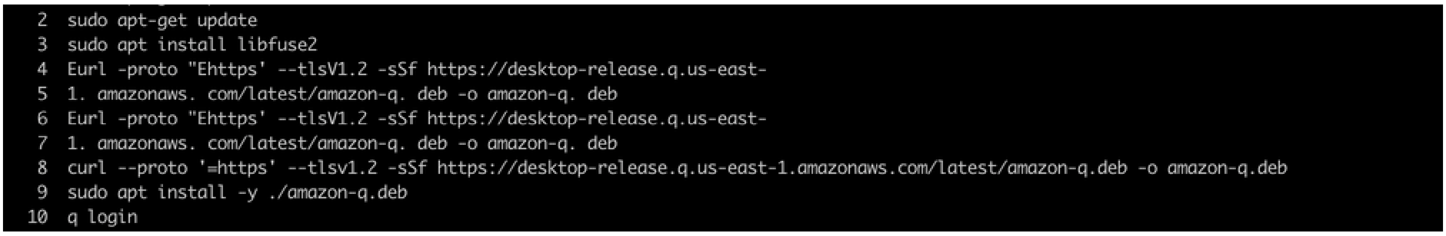
Free tier: In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where t2.micro isn't available) when used with free tier AMIs, 750 hours per month of public IPv4 address usage, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.



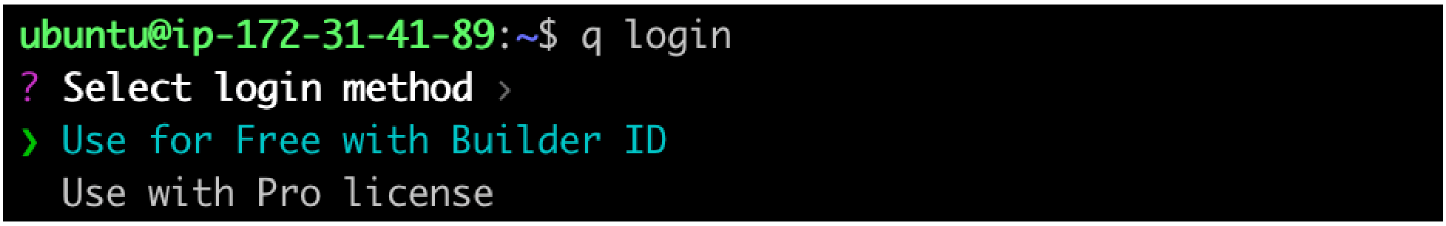
Instance type

Instance type

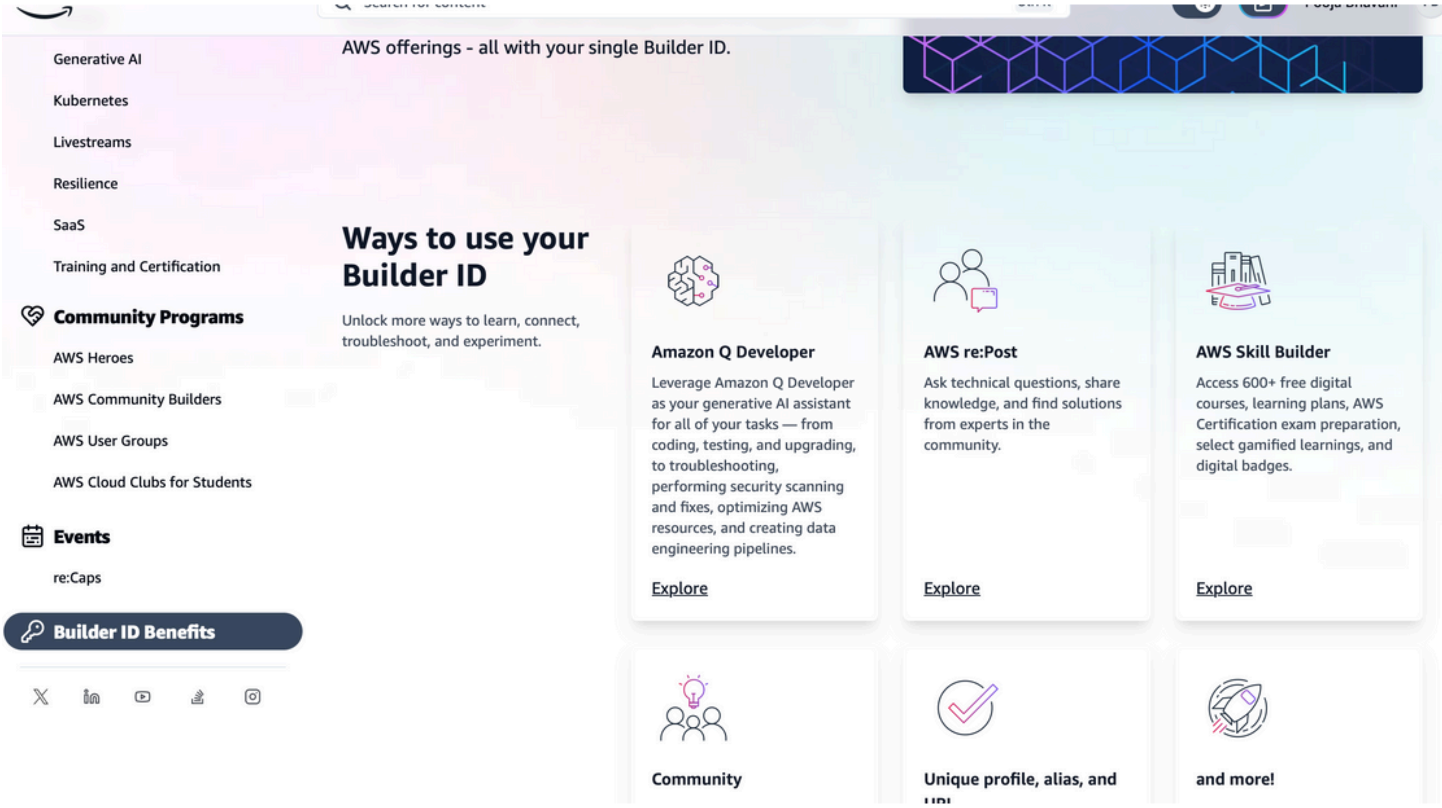
t2.micro — Free tier eligible
Family: t2 1 vCPU 1 GiB Memory Current generation: true
On-Demand Linux base pricing: 0.0124 USD per Hour
On-Demand Windows base pricing: 0.017 USD per Hour
On-Demand RHEL base pricing: 0.0268 USD per Hour
On-Demand Ubuntu Pro base pricing: 0.0142 USD per Hour
On-Demand SUSE base pricing: 0.0124 USD per Hour



```
 2  sudo apt-get update
 3  sudo apt install libfuse2
 4  Eurl -proto "Ehttps' --tlsV1.2 -sSf https://desktop-release.q.us-east-
 5  1. amazonaws. com/latest/amazon-q. deb -o amazon-q. deb
 6  Eurl -proto "Ehttps' --tlsV1.2 -sSf https://desktop-release.q.us-east-
 7  1. amazonaws. com/latest/amazon-q. deb -o amazon-q. deb
 8  curl --proto '=https' --tlsv1.2 -sSf https://desktop-release.q.us-east-1.amazonaws.com/latest/amazon-q.deb -o amazon-q.deb
 9  sudo apt install -y ./amazon-q.deb
10  q login
```



```
ubuntu@ip-172-31-41-89:~$ q login
? Select login method ›
❯ Use for Free with Builder ID
  Use with Pro license
```



AWS offerings - all with your single Builder ID.

Ways to use your Builder ID

Unlock more ways to learn, connect, troubleshoot, and experiment.

Amazon Q Developer

Leverage Amazon Q Developer as your generative AI assistant for all of your tasks — from coding, testing, and upgrading, to troubleshooting, performing security scanning and fixes, optimizing AWS resources, and creating data engineering pipelines.

Explore

AWS re:Post

Ask technical questions, share knowledge, and find solutions from experts in the community.

Explore

AWS Skill Builder

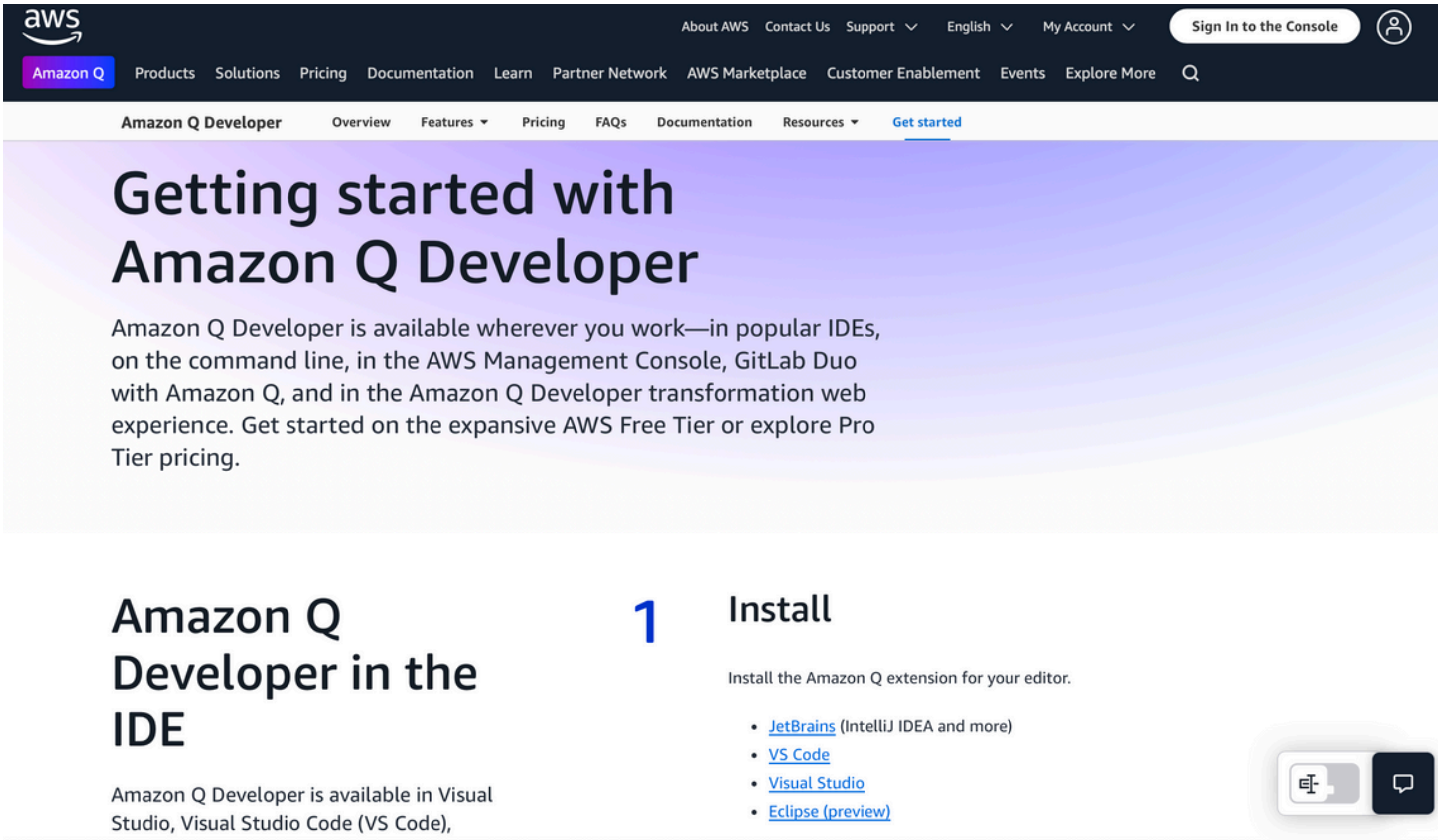
Access 600+ free digital courses, learning plans, AWS Certification exam preparation, select gamified learnings, and digital badges.

Explore

Community

Unique profile, alias, and

and more!



Getting started with Amazon Q Developer

Amazon Q Developer is available wherever you work—in popular IDEs, on the command line, in the AWS Management Console, GitLab Duo with Amazon Q, and in the Amazon Q Developer transformation web experience. Get started on the expansive AWS Free Tier or explore Pro Tier pricing.

Amazon Q Developer in the IDE

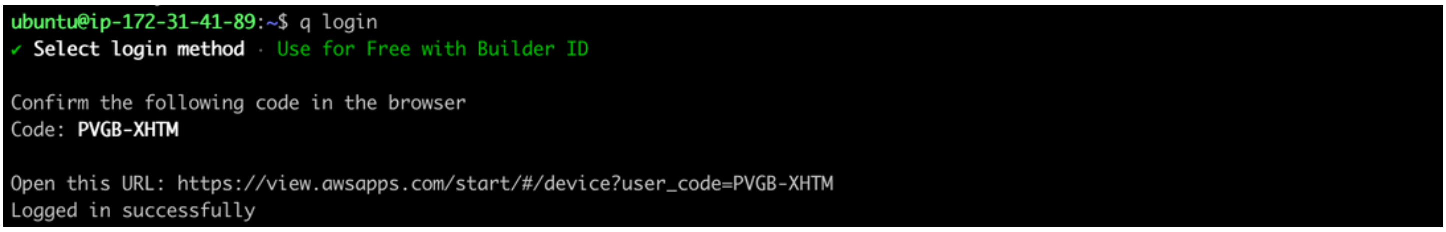
Amazon Q Developer is available in Visual Studio, Visual Studio Code (VS Code),

1

Install

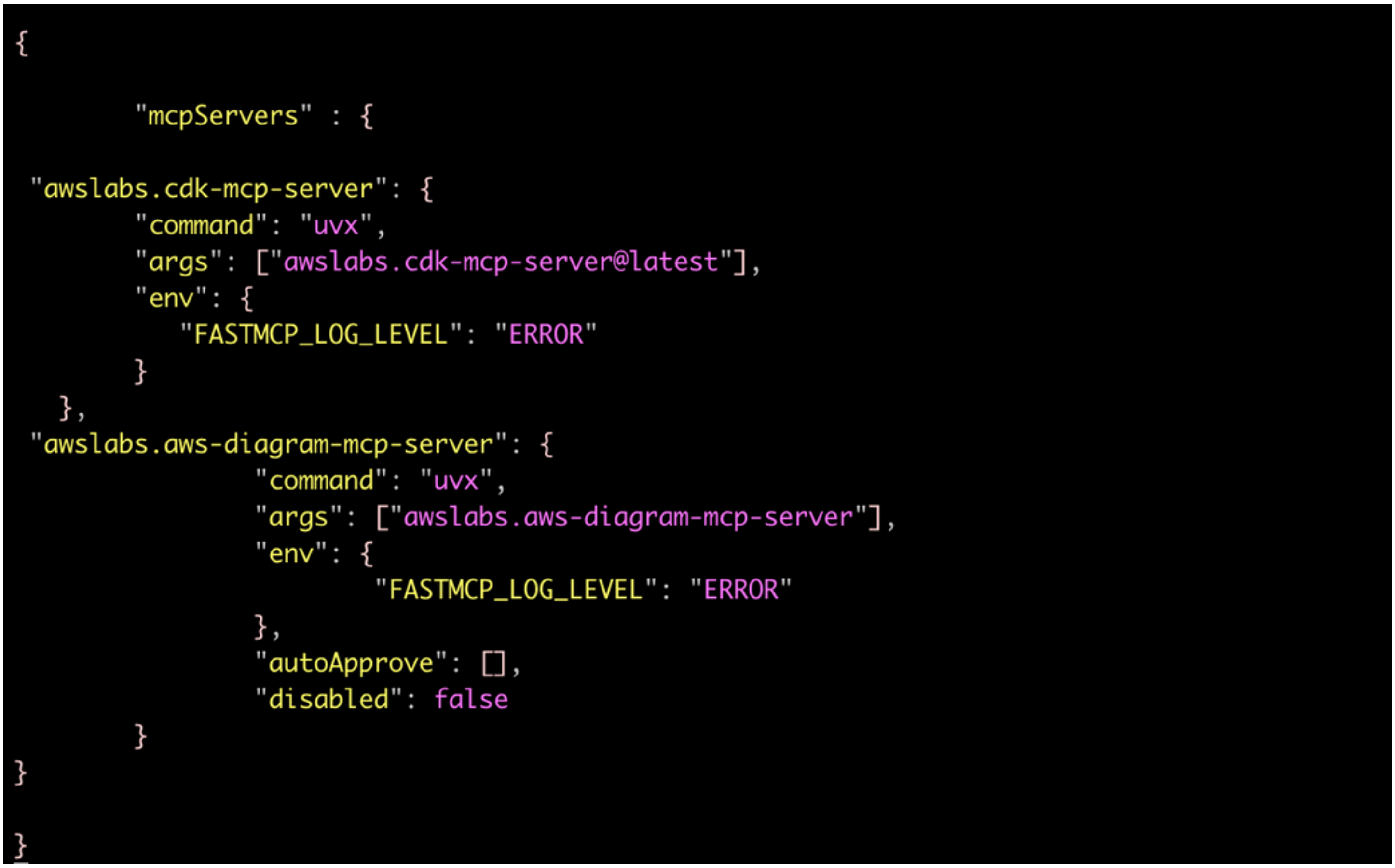
Install the Amazon Q extension for your editor.

- JetBrains (IntelliJ IDEA and more)
- VS Code
- Visual Studio
- Eclipse (preview)

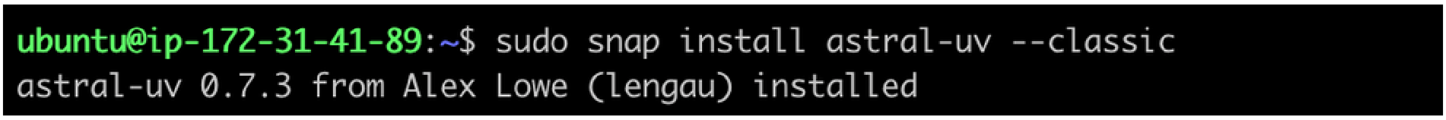


```
ubuntu@ip-172-31-41-89:~$ q login
✔ Select login method · Use for Free with Builder ID

Confirm the following code in the browser
Code: PVGB-XHTM

Open this URL: https://view.awsapps.com/start/#/device?user_code=PVGB-XHTM
Logged in successfully
```

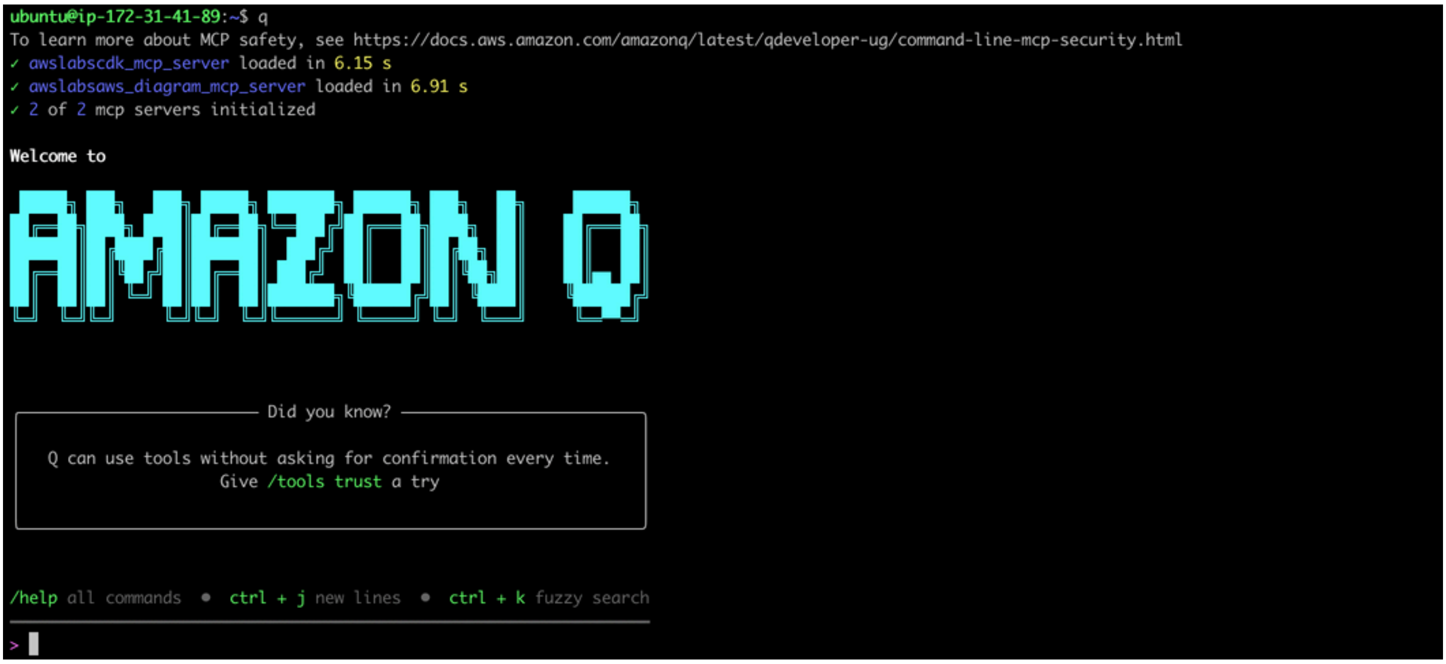
vim mcp.json



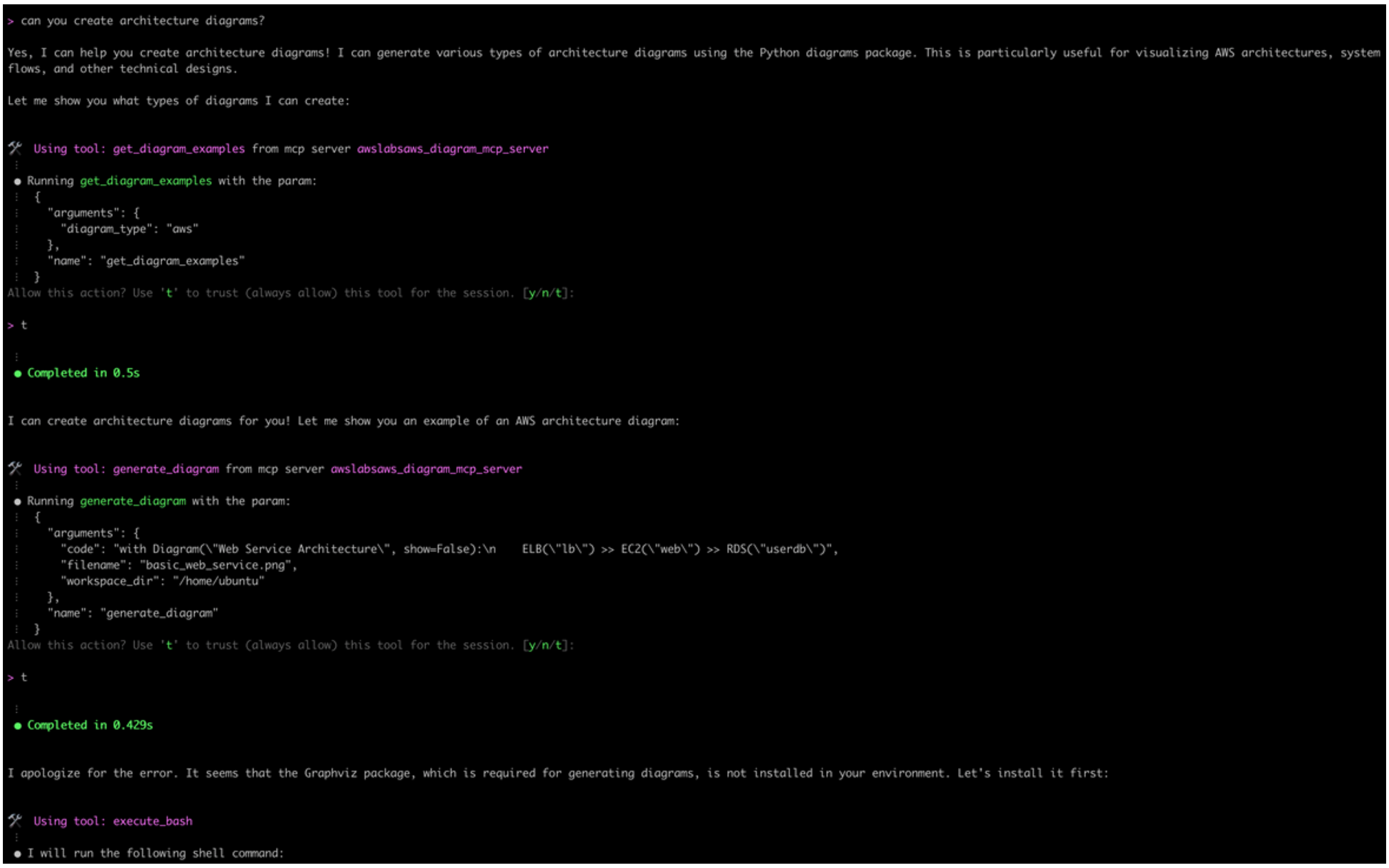
```
{

        "mcpServers" : {

 "awslabs.cdk-mcp-server": {
        "command": "uvx",
        "args": ["awslabs.cdk-mcp-server@latest"],
        "env": {
           "FASTMCP_LOG_LEVEL": "ERROR"
        }
   },
 "awslabs.aws-diagram-mcp-server": {
                "command": "uvx",
                "args": ["awslabs.aws-diagram-mcp-server"],
                "env": {
                        "FASTMCP_LOG_LEVEL": "ERROR"
                },
                "autoApprove": [],
                "disabled": false
        }
}

}
```



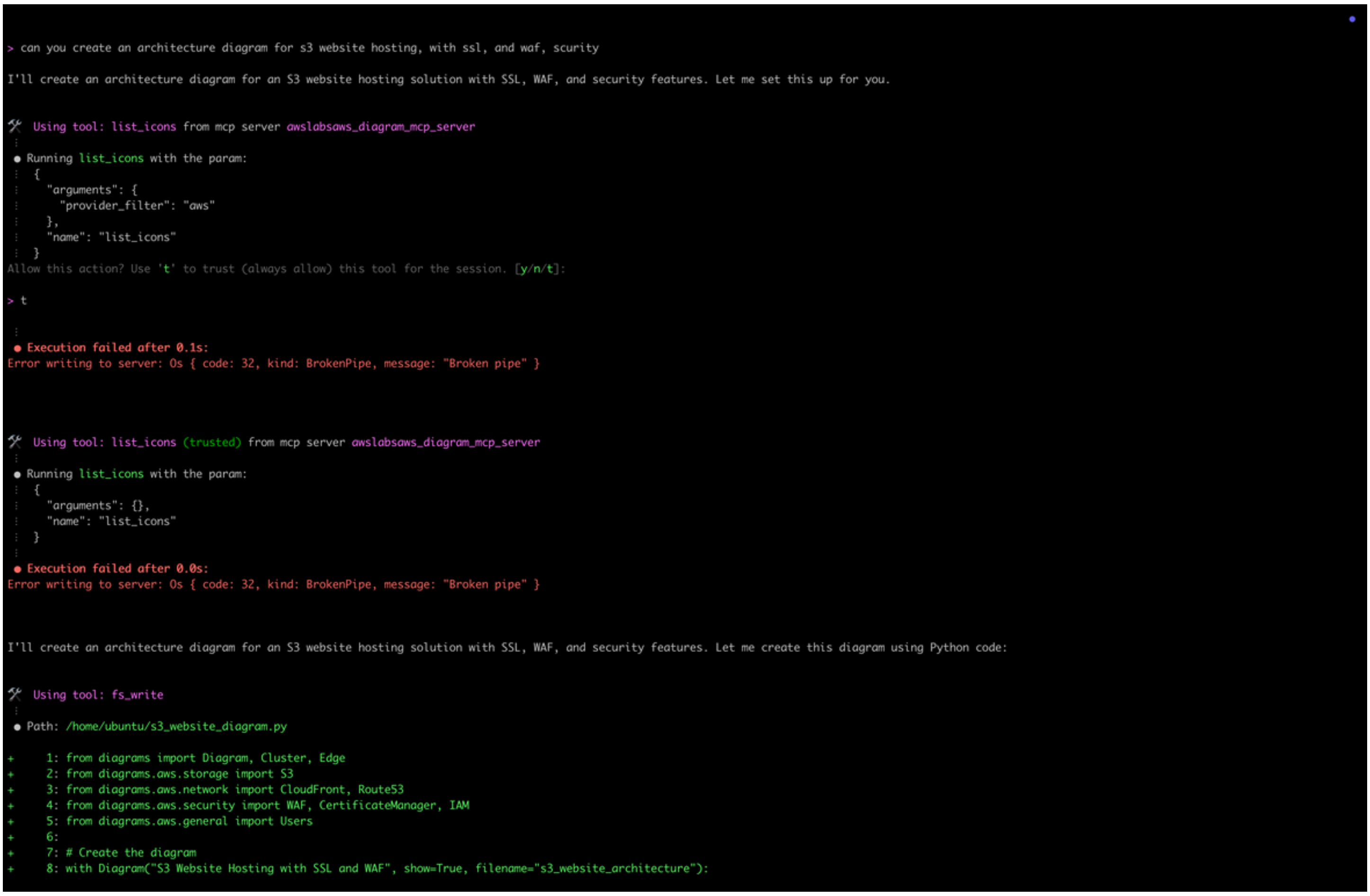
```
ubuntu@ip-172-31-41-89:~$ sudo snap install astral-uv --classic
astral-uv 0.7.3 from Alex Lowe (lengau) installed
```



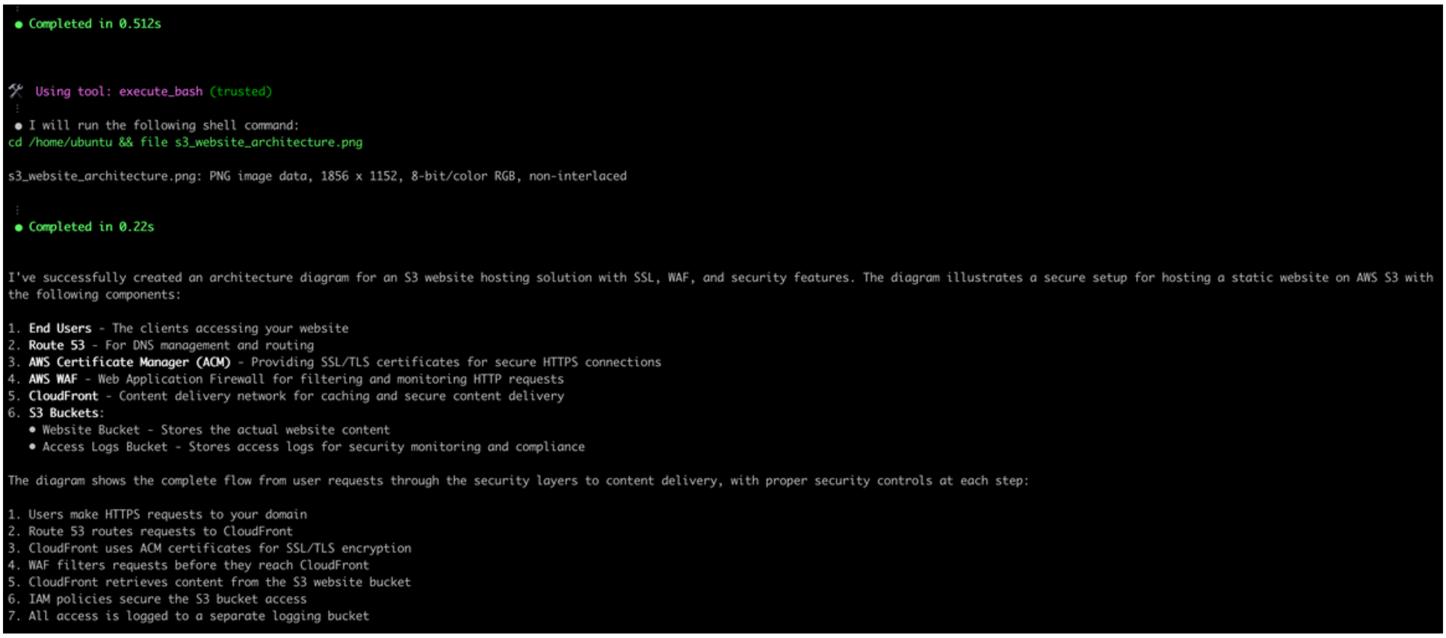
```
ubuntu@ip-172-31-41-89:~$ q
To learn more about MCP safety, see https://docs.aws.amazon.com/amazonq/latest/qdeveloper-ug/command-line-mcp-security.html
✓ awslabscdk_mcp_server loaded in 6.15 s
✓ awslabsaws_diagram_mcp_server loaded in 6.91 s
✓ 2 of 2 mcp servers initialized

Welcome to
AMAZON Q

 ─────────────────── Did you know? ───────────────────

     Q can use tools without asking for confirmation every time.
                  Give /tools trust a try


/help all commands  •  ctrl + j new lines  •  ctrl + k fuzzy search
──────────────────────────────────────────────────────────────────
> ▌
```



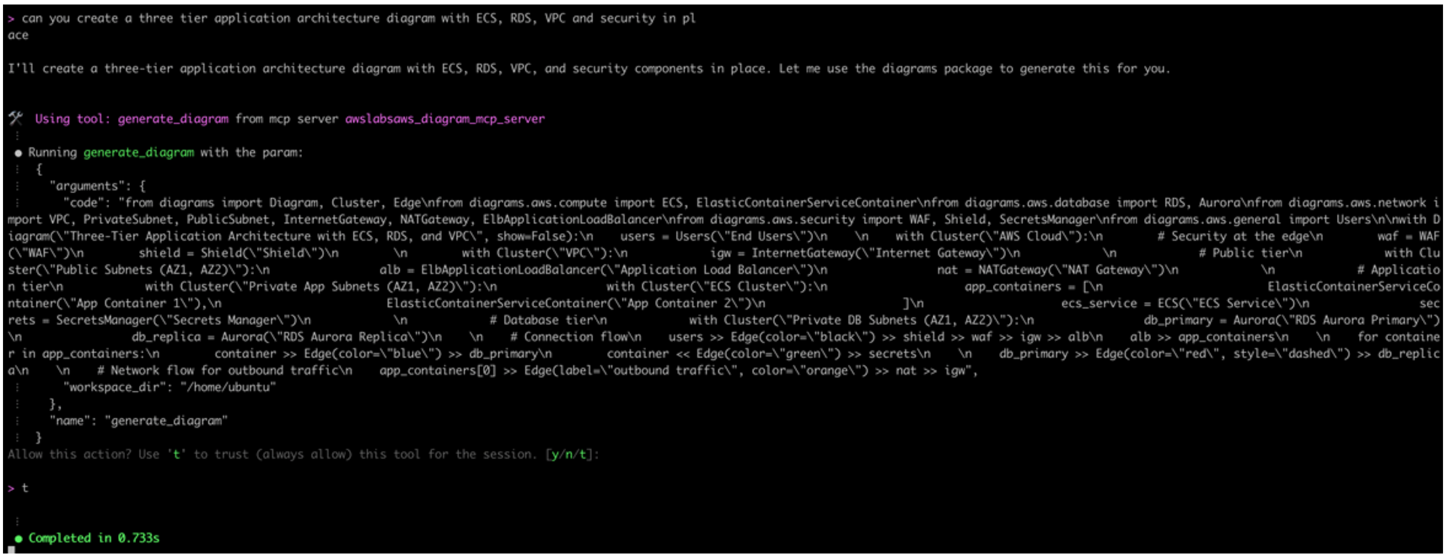
```
> can you create architecture diagrams?

Yes, I can help you create architecture diagrams! I can generate various types of architecture diagrams using the Python diagrams package. This is particularly useful for visualizing AWS architectures, system
flows, and other technical designs.

Let me show you what types of diagrams I can create:


🛠️  Using tool: get_diagram_examples from mcp server awslabsaws_diagram_mcp_server
 ⋮
 ● Running get_diagram_examples with the param:
 ⋮  {
 ⋮    "arguments": {
 ⋮      "diagram_type": "aws"
 ⋮    },
 ⋮    "name": "get_diagram_examples"
 ⋮  }
Allow this action? Use 't' to trust (always allow) this tool for the session. [y/n/t]:

> t

 ⋮
 ● Completed in 0.5s


I can create architecture diagrams for you! Let me show you an example of an AWS architecture diagram:


🛠️  Using tool: generate_diagram from mcp server awslabsaws_diagram_mcp_server
 ⋮
 ● Running generate_diagram with the param:
 ⋮  {
 ⋮    "arguments": {
 ⋮      "code": "with Diagram(\"Web Service Architecture\", show=False):\n    ELB(\"lb\") >> EC2(\"web\") >> RDS(\"userdb\")",
 ⋮      "filename": "basic_web_service.png",
 ⋮      "workspace_dir": "/home/ubuntu"
 ⋮    },
 ⋮    "name": "generate_diagram"
 ⋮  }
Allow this action? Use 't' to trust (always allow) this tool for the session. [y/n/t]:

> t

 ⋮
 ● Completed in 0.429s


I apologize for the error. It seems that the Graphviz package, which is required for generating diagrams, is not installed in your environment. Let's install it first:


🛠️  Using tool: execute_bash
 ⋮
 ● I will run the following shell command:
```



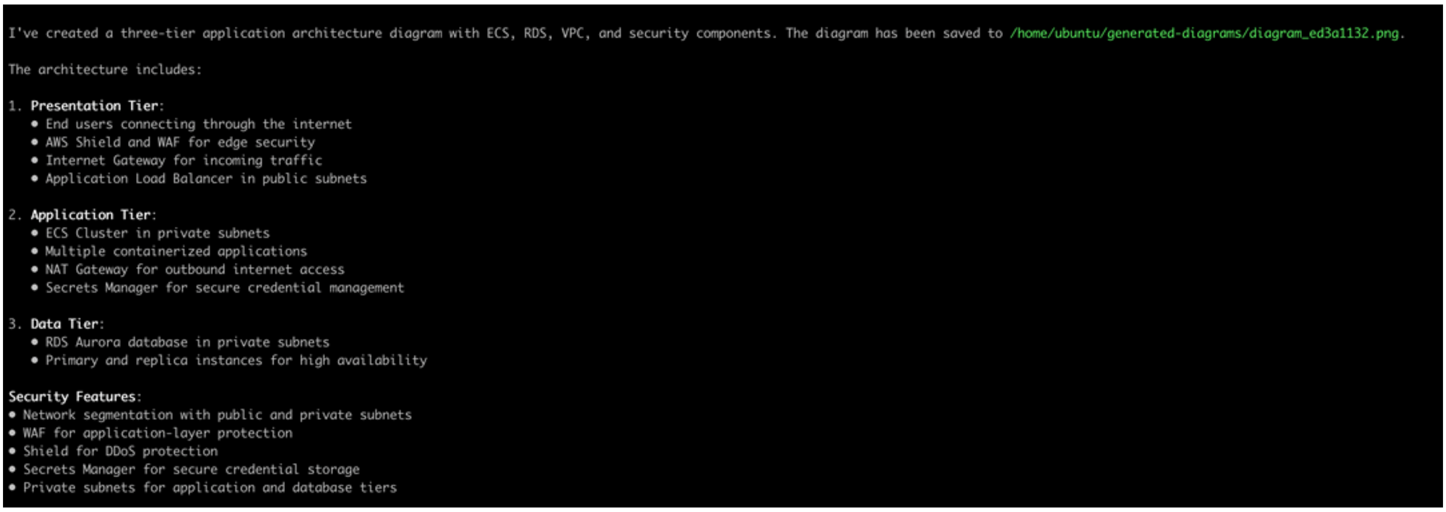
```
> can you create an architecture diagram for s3 website hosting, with ssl, and waf, scurity

I'll create an architecture diagram for an S3 website hosting solution with SSL, WAF, and security features. Let me set this up for you.


🛠️  Using tool: list_icons from mcp server awslabsaws_diagram_mcp_server
 ⋮
 ● Running list_icons with the param:
 ⋮  {
 ⋮    "arguments": {
 ⋮      "provider_filter": "aws"
 ⋮    },
 ⋮    "name": "list_icons"
 ⋮  }
Allow this action? Use 't' to trust (always allow) this tool for the session. [y/n/t]:

> t

 ⋮
 ● Execution failed after 0.1s:
Error writing to server: Os { code: 32, kind: BrokenPipe, message: "Broken pipe" }



🛠️  Using tool: list_icons (trusted) from mcp server awslabsaws_diagram_mcp_server
 ⋮
 ● Running list_icons with the param:
 ⋮  {
 ⋮    "arguments": {},
 ⋮    "name": "list_icons"
 ⋮  }
 ⋮
 ● Execution failed after 0.0s:
Error writing to server: Os { code: 32, kind: BrokenPipe, message: "Broken pipe" }



I'll create an architecture diagram for an S3 website hosting solution with SSL, WAF, and security features. Let me create this diagram using Python code:


🛠️  Using tool: fs_write
 ⋮
 ● Path: /home/ubuntu/s3_website_diagram.py

+     1: from diagrams import Diagram, Cluster, Edge
+     2: from diagrams.aws.storage import S3
+     3: from diagrams.aws.network import CloudFront, Route53
+     4: from diagrams.aws.security import WAF, CertificateManager, IAM
+     5: from diagrams.aws.general import Users
+     6:
+     7: # Create the diagram
+     8: with Diagram("S3 Website Hosting with SSL and WAF", show=True, filename="s3_website_architecture"):
```



```
 ⋮
 ● Completed in 0.512s


🛠️  Using tool: execute_bash (trusted)
 ⋮
 ● I will run the following shell command:
cd /home/ubuntu && file s3_website_architecture.png

s3_website_architecture.png: PNG image data, 1856 x 1152, 8-bit/color RGB, non-interlaced

 ⋮
 ● Completed in 0.22s

I've successfully created an architecture diagram for an S3 website hosting solution with SSL, WAF, and security features. The diagram illustrates a secure setup for hosting a static website on AWS S3 with
the following components:

1. End Users - The clients accessing your website
2. Route 53 - For DNS management and routing
3. AWS Certificate Manager (ACM) - Providing SSL/TLS certificates for secure HTTPS connections
4. AWS WAF - Web Application Firewall for filtering and monitoring HTTP requests
5. CloudFront - Content delivery network for caching and secure content delivery
6. S3 Buckets:
   • Website Bucket - Stores the actual website content
   • Access Logs Bucket - Stores access logs for security monitoring and compliance

The diagram shows the complete flow from user requests through the security layers to content delivery, with proper security controls at each step:

1. Users make HTTPS requests to your domain
2. Route 53 routes requests to CloudFront
3. CloudFront uses ACM certificates for SSL/TLS encryption
4. WAF filters requests before they reach CloudFront
5. CloudFront retrieves content from the S3 website bucket
6. IAM policies secure the S3 bucket access
7. All access is logged to a separate logging bucket
```



```
> can you create a three tier application architecture diagram with ECS, RDS, VPC and security in pl
ace

I'll create a three-tier application architecture diagram with ECS, RDS, VPC, and security components in place. Let me use the diagrams package to generate this for you.


🛠️  Using tool: generate_diagram from mcp server awslabsaws_diagram_mcp_server
 ⋮
 ● Running generate_diagram with the param:
 ⋮  {
 ⋮    "arguments": {
 ⋮      "code": "from diagrams import Diagram, Cluster, Edge\nfrom diagrams.aws.compute import ECS, ElasticContainerServiceContainer\nfrom diagrams.aws.database import RDS, Aurora\nfrom diagrams.aws.network i
mport VPC, PrivateSubnet, PublicSubnet, InternetGateway, NATGateway, ElbApplicationLoadBalancer\nfrom diagrams.aws.security import WAF, Shield, SecretsManager\nfrom diagrams.aws.general import Users\n\nwith D
iagram(\"Three-Tier Application Architecture with ECS, RDS, and VPC\", show=False):\n    users = Users(\"End Users\")\n    \n    with Cluster(\"AWS Cloud\"):\n        # Security at the edge\n        waf = WAF
(\"WAF\")\n        shield = Shield(\"Shield\")\n        \n        with Cluster(\"VPC\"):\n            igw = InternetGateway(\"Internet Gateway\")\n            \n            # Public tier\n            with Clu
ster(\"Public Subnets (AZ1, AZ2)\"):\n                alb = ElbApplicationLoadBalancer(\"Application Load Balancer\")\n                nat = NATGateway(\"NAT Gateway\")\n            \n            # Applicatio
n tier\n            with Cluster(\"Private App Subnets (AZ1, AZ2)\"):\n                with Cluster(\"ECS Cluster\"):\n                    app_containers = [\n                        ElasticContainerServiceCo
ntainer(\"App Container 1\"),\n                        ElasticContainerServiceContainer(\"App Container 2\")\n                    ]\n                    ecs_service = ECS(\"ECS Service\")\n                sec
rets = SecretsManager(\"Secrets Manager\")\n            \n            # Database tier\n            with Cluster(\"Private DB Subnets (AZ1, AZ2)\"):\n                db_primary = Aurora(\"RDS Aurora Primary\")
\n                db_replica = Aurora(\"RDS Aurora Replica\")\n    \n    # Connection flow\n    users >> Edge(color=\"black\") >> shield >> waf >> igw >> alb\n    alb >> app_containers\n    \n    for containe
r in app_containers:\n        container >> Edge(color=\"blue\") >> db_primary\n        container << Edge(color=\"green\") >> secrets\n    \n    db_primary >> Edge(color=\"red\", style=\"dashed\") >> db_replic
a\n    \n    # Network flow for outbound traffic\n    app_containers[0] >> Edge(label=\"outbound traffic\", color=\"orange\") >> nat >> igw",
 ⋮      "workspace_dir": "/home/ubuntu"
 ⋮    },
 ⋮    "name": "generate_diagram"
 ⋮  }
Allow this action? Use 't' to trust (always allow) this tool for the session. [y/n/t]:

> t

 ⋮
 ● Completed in 0.733s
```



```
I've created a three-tier application architecture diagram with ECS, RDS, VPC, and security components. The diagram has been saved to /home/ubuntu/generated-diagrams/diagram_ed3a1132.png.

The architecture includes:

1. Presentation Tier:
   • End users connecting through the internet
   • AWS Shield and WAF for edge security
   • Internet Gateway for incoming traffic
   • Application Load Balancer in public subnets

2. Application Tier:
   • ECS Cluster in private subnets
   • Multiple containerized applications
   • NAT Gateway for outbound internet access
   • Secrets Manager for secure credential management

3. Data Tier:
   • RDS Aurora database in private subnets
   • Primary and replica instances for high availability

Security Features:
• Network segmentation with public and private subnets
• WAF for application-layer protection
• Shield for DDoS protection
• Secrets Manager for secure credential storage
• Private subnets for application and database tiers
```