

AWS Tools Documentation

This document provides detailed information about the AWS tools available in the DevOps MCP Server.

Overview

The AWS integration allows you to interact with your AWS services through the MCP protocol. It provides tools for retrieving information about ECS, S3, ELB, VPC, and Cost Explorer.

Configuration

The AWS integration can be configured using the following environment variables:

- **AWS_REGION**: The AWS region to use (default: **us-east-1**)
- **AWS_ACCESS_KEY_ID**: Your AWS access key ID
- **AWS_SECRET_ACCESS_KEY**: Your AWS secret access key
- **AWS_SESSION_TOKEN**: Your AWS session token (optional)
- **AWS_TIMEOUT**: Timeout for API calls in seconds (default: **15**)

Available Tools

The AWS integration provides the following tools:

ECS Tools

list_ecs_clusters

List ECS clusters.

Parameters:

- **max_results** (optional): Maximum number of results to return (default: 100, max: 100)

Example:

```
list_ecs_clusters()  
list_ecs_clusters(50)
```

get_ecs_cluster

Get details of an ECS cluster.

Parameters:

- **cluster** (required): Cluster name or ARN

Example:

```
get_ecs_cluster("my-cluster")
```

list_ecs_services

List ECS services in a cluster.

Parameters:

- **cluster** (required): Cluster name or ARN
- **max_results** (optional): Maximum number of results to return (default: 100, max: 100)

Example:

```
list_ecs_services("my-cluster")  
list_ecs_services("my-cluster", 50)
```

get_ecs_service

Get details of an ECS service.

Parameters:

- **cluster** (required): Cluster name or ARN
- **service** (required): Service name or ARN

Example:

```
get_ecs_service("my-cluster", "my-service")
```

PROF

list_ecs_tasks

List ECS tasks in a cluster.

Parameters:

- **cluster** (required): Cluster name or ARN
- **service** (optional): Service name or ARN
- **max_results** (optional): Maximum number of results to return (default: 100, max: 100)

Example:

```
list_ecs_tasks("my-cluster")  
list_ecs_tasks("my-cluster", "my-service", 50)
```

get_ecs_task

Get details of an ECS task.

Parameters:

- **cluster** (required): Cluster name or ARN
- **task** (required): Task name or ARN

Example:

```
get_ecs_task("my-cluster", "my-task")
```

list_ecs_task_definitions

List ECS task definitions.

Parameters:

- **family_prefix** (optional): Family prefix to filter by
- **max_results** (optional): Maximum number of results to return (default: 100, max: 100)

Example:

```
list_ecs_task_definitions()  
list_ecs_task_definitions("my-family", 50)
```

get_ecs_task_definition

Get details of an ECS task definition.

Parameters:

- **task_definition** (required): Task definition name or ARN

Example:

```
get_ecs_task_definition("my-task-definition")
```

S3 Tools

list_s3_buckets

List S3 buckets.

Example:

```
list_s3_buckets()
```

get_s3_bucket

Get details of an S3 bucket.

Parameters:

- **bucket** (required): Bucket name

Example:

```
get_s3_bucket("my-bucket")
```

list_s3_objects

List objects in an S3 bucket.

Parameters:

- **bucket** (required): Bucket name
- **prefix** (optional): Object key prefix
- **max_keys** (optional): Maximum number of keys to return (default: 1000, max: 1000)

Example:

```
list_s3_objects("my-bucket")  
list_s3_objects("my-bucket", "folder/", 100)
```

PROF

get_s3_object

Get metadata of an S3 object.

Parameters:

- **bucket** (required): Bucket name
- **key** (required): Object key

Example:

```
get_s3_object("my-bucket", "folder/file.txt")
```

get_s3_object_url

Generate a presigned URL for an S3 object.

Parameters:

- **bucket** (required): Bucket name
- **key** (required): Object key
- **expires_in** (optional): URL expiration time in seconds (default: 3600, max: 604800)

Example:

```
get_s3_object_url("my-bucket", "folder/file.txt")
get_s3_object_url("my-bucket", "folder/file.txt", 7200)
```

get_s3_bucket_size

Get the size of an S3 bucket.

Parameters:

- **bucket** (required): Bucket name
- **prefix** (optional): Object key prefix

Example:

```
get_s3_bucket_size("my-bucket")
get_s3_bucket_size("my-bucket", "folder/")
```

ELB Tools

PROF

list_load_balancers

List Application Load Balancers.

Parameters:

- **max_results** (optional): Maximum number of results to return (default: 100, max: 100)

Example:

```
list_load_balancers()
list_load_balancers(50)
```

get_load_balancer

Get details of a load balancer.

Parameters:

- `load_balancer_arn` (required): Load balancer ARN

Example:

```
get_load_balancer("arn:aws:elasticloadbalancing:us-east-1:123456789012:loadbalancer/app/my-load-balancer/1234567890abcdef")
```

`list_listeners`

List listeners for a load balancer.

Parameters:

- `load_balancer_arn` (required): Load balancer ARN

Example:

```
list_listeners("arn:aws:elasticloadbalancing:us-east-1:123456789012:loadbalancer/app/my-load-balancer/1234567890abcdef")
```

`get_listener`

Get details of a listener.

Parameters:

- `listener_arn` (required): Listener ARN

Example:

```
get_listener("arn:aws:elasticloadbalancing:us-east-1:123456789012:listener/app/my-load-balancer/1234567890abcdef/1234567890abcdef")
```

`list_target_groups`

List target groups.

Parameters:

- `load_balancer_arn` (optional): Load balancer ARN
- `max_results` (optional): Maximum number of results to return (default: 100, max: 100)

Example:

```
list_target_groups()  
list_target_groups("arn:aws:elasticloadbalancing:us-east-  
1:123456789012:loadbalancer/app/my-load-balancer/1234567890abcdef", 50)
```

get_target_group

Get details of a target group.

Parameters:

- **target_group_arn** (required): Target group ARN

Example:

```
get_target_group("arn:aws:elasticloadbalancing:us-east-  
1:123456789012:targetgroup/my-target-group/1234567890abcdef")
```

list_target_health

List health of targets in a target group.

Parameters:

- **target_group_arn** (required): Target group ARN

Example:

```
list_target_health("arn:aws:elasticloadbalancing:us-east-  
1:123456789012:targetgroup/my-target-group/1234567890abcdef")
```

list_rules

List rules for a listener.

Parameters:

- **listener_arn** (required): Listener ARN

Example:

```
list_rules("arn:aws:elasticloadbalancing:us-east-  
1:123456789012:listener/app/my-load-  
balancer/1234567890abcdef/1234567890abcdef")
```

VPC Tools

`list_vpcs`

List VPCs.

Parameters:

- `max_results` (optional): Maximum number of results to return (default: 100, max: 1000)

Example:

```
list_vpcs()  
list_vpcs(50)
```

`get_vpc`

Get details of a VPC.

Parameters:

- `vpc_id` (required): VPC ID

Example:

```
get_vpc("vpc-1234567890abcdef0")
```

`list_subnets`

List subnets.

Parameters:

- `vpc_id` (optional): VPC ID
- `max_results` (optional): Maximum number of results to return (default: 100, max: 1000)

Example:

```
list_subnets()  
list_subnets("vpc-1234567890abcdef0", 50)
```

`get_subnet`

Get details of a subnet.

Parameters:

- **subnet_id** (required): Subnet ID

Example:

```
get_subnet("subnet-1234567890abcdef0")
```

list_security_groups

List security groups.

Parameters:

- **vpc_id** (optional): VPC ID
- **max_results** (optional): Maximum number of results to return (default: 100, max: 1000)

Example:

```
list_security_groups()  
list_security_groups("vpc-1234567890abcdef0", 50)
```

get_security_group

Get details of a security group.

Parameters:

- **security_group_id** (required): Security group ID

Example:

```
get_security_group("sg-1234567890abcdef0")
```

list_route_tables

List route tables.

Parameters:

- **vpc_id** (optional): VPC ID
- **max_results** (optional): Maximum number of results to return (default: 100, max: 1000)

Example:

```
list_route_tables()  
list_route_tables("vpc-1234567890abcdef0", 50)
```

get_route_table

Get details of a route table.

Parameters:

- **route_table_id** (required): Route table ID

Example:

```
get_route_table("rtb-1234567890abcdef0")
```

Cost Explorer Tools

get_aws_cost_and_usage

Get AWS cost and usage data.

Parameters:

- **start_date** (optional): Start date in YYYY-MM-DD format (default: 30 days ago)
- **end_date** (optional): End date in YYYY-MM-DD format (default: today)
- **granularity** (optional): Time granularity (DAILY, MONTHLY, or HOURLY) (default: MONTHLY)
- **metrics** (optional): Comma-separated list of cost metrics to return (default: "BlendedCost,UnblendedCost,UsageQuantity")

Example:

```
get_aws_cost_and_usage()  
get_aws_cost_and_usage("2023-01-01", "2023-01-31", "DAILY",  
"BlendedCost")
```

get_aws_cost_by_service

Get AWS cost data grouped by service.

Parameters:

- **start_date** (optional): Start date in YYYY-MM-DD format (default: 30 days ago)
- **end_date** (optional): End date in YYYY-MM-DD format (default: today)
- **granularity** (optional): Time granularity (DAILY, MONTHLY, or HOURLY) (default: MONTHLY)

Example:

```
get_aws_cost_by_service()  
get_aws_cost_by_service("2023-01-01", "2023-01-31", "DAILY")
```

get_aws_cost_by_account

Get AWS cost data grouped by account.

Parameters:

- **start_date** (optional): Start date in YYYY-MM-DD format (default: 30 days ago)
- **end_date** (optional): End date in YYYY-MM-DD format (default: today)
- **granularity** (optional): Time granularity (DAILY, MONTHLY, or HOURLY) (default: MONTHLY)

Example:

```
get_aws_cost_by_account()  
get_aws_cost_by_account("2023-01-01", "2023-01-31", "DAILY")
```

get_aws_cost_by_region

Get AWS cost data grouped by region.

Parameters:

- **start_date** (optional): Start date in YYYY-MM-DD format (default: 30 days ago)
- **end_date** (optional): End date in YYYY-MM-DD format (default: today)
- **granularity** (optional): Time granularity (DAILY, MONTHLY, or HOURLY) (default: MONTHLY)

Example:

```
get_aws_cost_by_region()  
get_aws_cost_by_region("2023-01-01", "2023-01-31", "DAILY")
```

get_aws_cost_forecast

Get AWS cost forecast.

Parameters:

- **start_date** (optional): Start date in YYYY-MM-DD format (default: today)
- **end_date** (optional): End date in YYYY-MM-DD format (default: 30 days from now)
- **granularity** (optional): Time granularity (DAILY, MONTHLY, or HOURLY) (default: MONTHLY)

- **metric** (optional): Cost metric to forecast (BLENDED_COST, UNBLENDED_COST, or AMORTIZED_COST) (default: BLENDED_COST)

Example:

```
get_aws_cost_forecast()  
get_aws_cost_forecast("2023-01-01", "2023-01-31", "DAILY",  
"UNBLENDED_COST")
```

get_aws_cost_anomalies

Get AWS cost anomalies.

Parameters:

- **start_date** (optional): Start date in YYYY-MM-DD format (default: 30 days ago)
- **end_date** (optional): End date in YYYY-MM-DD format (default: today)
- **max_results** (optional): Maximum number of results to return (default: 100, max: 100)

Example:

```
get_aws_cost_anomalies()  
get_aws_cost_anomalies("2023-01-01", "2023-01-31", 50)
```

Available Resources

The AWS integration provides the following resources:

ECS Resources

- **aws://ecs/clusters**: List all ECS clusters
- **aws://ecs/cluster/{cluster}**: Get details of an ECS cluster
- **aws://ecs/cluster/{cluster}/services**: List all services in an ECS cluster
- **aws://ecs/cluster/{cluster}/service/{service}**: Get details of an ECS service
- **aws://ecs/cluster/{cluster}/tasks**: List all tasks in an ECS cluster

S3 Resources

- **aws://s3/buckets**: List all S3 buckets
- **aws://s3/bucket/{bucket}**: Get details of an S3 bucket
- **aws://s3/bucket/{bucket}/objects/{prefix}**: List objects in an S3 bucket with optional prefix

ELB Resources

- **aws://elb/load-balancers**: List all load balancers

- `aws://elb/load-balancer/{loadBalancerArn}`: Get details of a load balancer
- `aws://elb/load-balancer/{loadBalancerArn}/listeners`: List all listeners for a load balancer
- `aws://elb/load-balancer/{loadBalancerArn}/target-groups`: List all target groups for a load balancer

VPC Resources

- `aws://vpc/vpcs`: List all VPCs
- `aws://vpc/vpc/{vpcId}`: Get details of a VPC
- `aws://vpc/vpc/{vpcId}/subnets`: List all subnets in a VPC
- `aws://vpc/vpc/{vpcId}/security-groups`: List all security groups in a VPC
- `aws://vpc/vpc/{vpcId}/route-tables`: List all route tables in a VPC

Cost Resources

- `aws://cost/by-service`: Get AWS cost data grouped by service
- `aws://cost/by-account`: Get AWS cost data grouped by account
- `aws://cost/by-region`: Get AWS cost data grouped by region
- `aws://cost/forecast`: Get AWS cost forecast

Examples

Working with ECS

```
# List all ECS clusters
list_ecs_clusters()

# Get details of a cluster
get_ecs_cluster("my-cluster")

# List all services in a cluster
list_ecs_services("my-cluster")

# Get details of a service
get_ecs_service("my-cluster", "my-service")

# List all tasks in a cluster
list_ecs_tasks("my-cluster")
```

Working with S3

```
# List all S3 buckets
list_s3_buckets()

# Get details of a bucket
get_s3_bucket("my-bucket")
```

```
# List objects in a bucket
list_s3_objects("my-bucket", "folder/")

# Get metadata of an object
get_s3_object("my-bucket", "folder/file.txt")

# Generate a presigned URL for an object
get_s3_object_url("my-bucket", "folder/file.txt")
```

Working with VPC

```
# List all VPCs
list_vpcs()

# Get details of a VPC
get_vpc("vpc-1234567890abcdef0")

# List all subnets in a VPC
list_subnets("vpc-1234567890abcdef0")

# List all security groups in a VPC
list_security_groups("vpc-1234567890abcdef0")
```

Working with Cost Explorer

```
# Get cost data grouped by service
get_aws_cost_by_service()

# Get cost data grouped by region
get_aws_cost_by_region()

# Get cost forecast
get_aws_cost_forecast()
```