

Welcome!

SPRINGFIELD AMAZON WEB SERVICES USER GROUP
JULY 2021

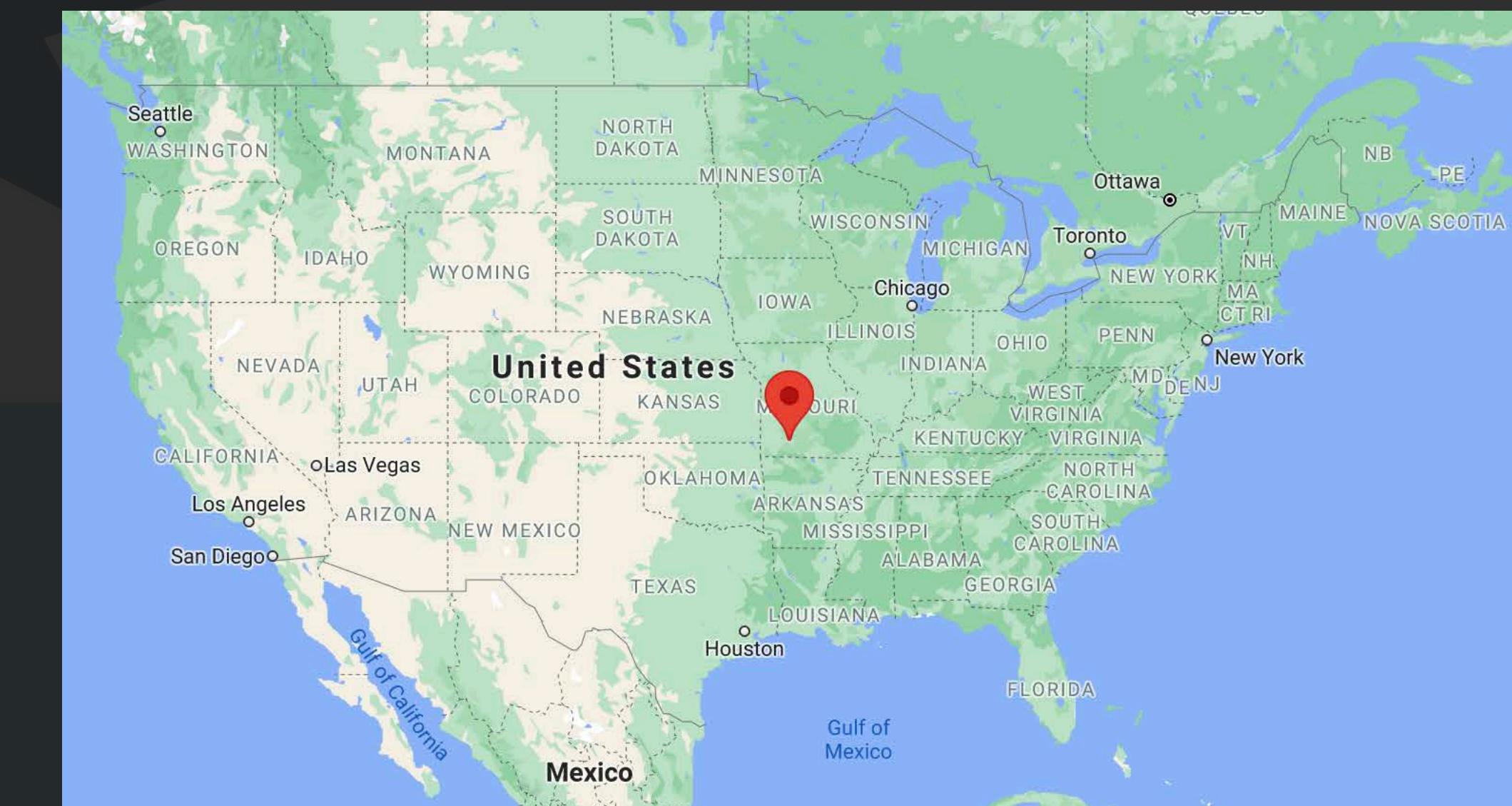
#SGFAWS

INTRO TO AWS CLI AND SDK: USING COMMAND LINE & DEV TOOLS TO MANAGE AWS

ABOUT SPRINGFIELD AWS

- ▶ **Meetup**
<https://meetup.com/sgfaws/>
- ▶ **YouTube**
<https://youtube.com/c/sgfaws>
- ▶ **Discord**
<https://sgf.dev/>

Springfield Amazon Web Services (SGF AWS) User Group is a community-based user group that promotes and advocates for Amazon Web Services in the Springfield Missouri region.

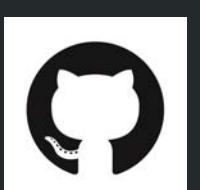


ABOUT JASON KLEIN

- ▶ **20+ years experience in IT (Networks and Linux) and Software Development (PHP/MySQL)**
- ▶ I have been an AWS user for 8+ years. First for DR, then forklift moved our Linux servers, before updating platform to use native AWS services (S3, RDS, SQS, SNS, SES, ECS/ECR, CP/CP (CI/CD)...
- ▶ I have been using CLI and APIs to manage various hosting infrastructure for nearly 20 years.
- ▶ Follow me!



@JasnK



@jason-klein



AGENDA

- ▶ Terminology - AWS Management Tools
- ▶ CLI - Benefits, Prerequisites, Examples
- ▶ SDK - Benefits, Prerequisites, Examples
- ▶ Resources

TERMINOLOGY - AWS MANAGEMENT TOOLS

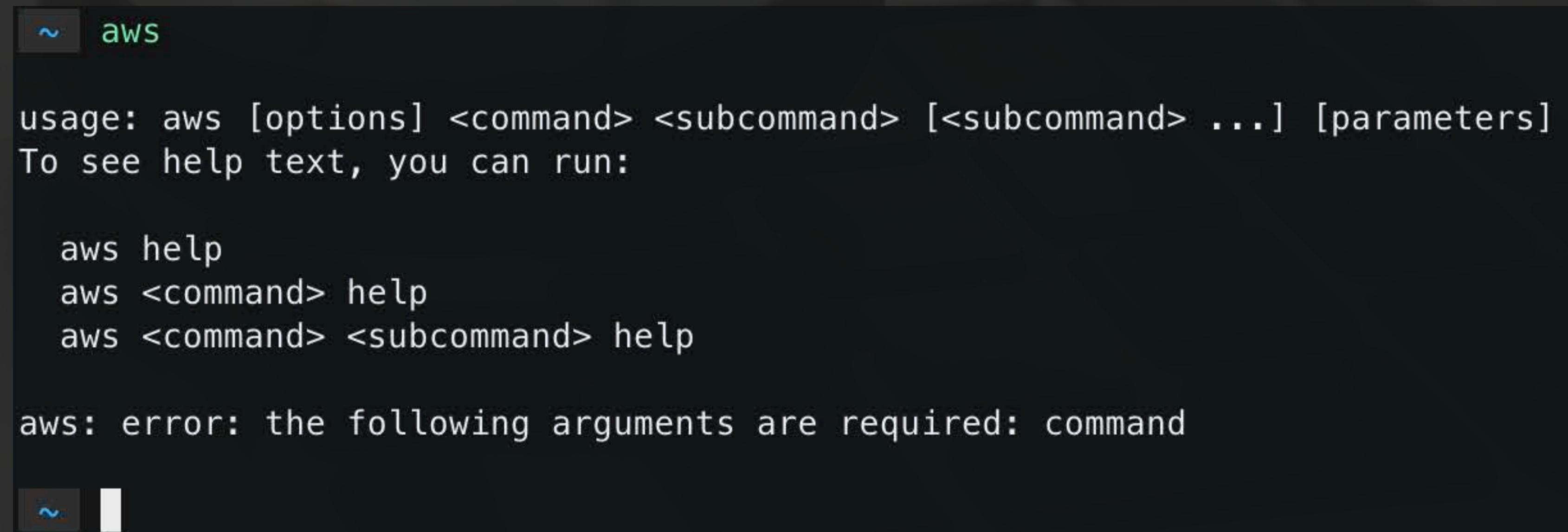
- ▶ Web Console <https://aws.amazon.com/console/>
- ▶ CloudFormation <https://aws.amazon.com/cloudformation/>
- ▶ API (Application Programming Interface) <https://docs.aws.amazon.com/>
- ▶ CLI (Command Line Interface) <https://docs.aws.amazon.com/cli/>
- ▶ SDK (Software Development Kit) <https://aws.amazon.com/tools/>
- ▶ SAM CLI (Serverless Application Model) <https://aws.amazon.com/serverless/sam/>
- ▶ CDK (Cloud Development Kit) <https://docs.aws.amazon.com/cdk/>

TERMINOLOGY - AWS MANAGEMENT TOOLS

- ▶ Web Console <https://aws.amazon.com/console/>
- ▶ CloudFormation <https://aws.amazon.com/cloudformation/>
- ▶ API (**A**pplication **P**rogramming **I**nterface) <https://docs.aws.amazon.com/>
- ▶ CLI (**C**ommand **L**ine **I**nterface) <https://docs.aws.amazon.com/cli/>
- ▶ SDK (**S**oftware **D**evelopment **K**it) <https://aws.amazon.com/tools/>
- ▶ SAM CLI (**S**erverless **A**pplication **M**odel) <https://aws.amazon.com/serverless/sam/>
- ▶ CDK (**C**loud **D**evelopment **K**it) <https://docs.aws.amazon.com/cdk/>

CLI BENEFITS

- ▶ Control AWS services through Command Line
- ▶ Automate infrastructure through scripts
- ▶ Faster and more efficient than AWS web console, especially for frequent or complex tasks



```
~ aws
usage: aws [options] <command> <subcommand> [<subcommand> ...] [parameters]
To see help text, you can run:

aws help
aws <command> help
aws <command> <subcommand> help

aws: error: the following arguments are required: command
```

CLI FEATURES

- ▶ 267 top-level commands
- ▶ 6000+ total commands
- ▶ 37,000+ total parameters

▶ accessanalyzer	codebuild	ec2	importexport	machinelearning	qldb-session	sqs
▶ acm	codecommit	ec2-instance-connect	inspector	macie	quicksight	ssm
▶ acm-pca	codeguru-reviewer	ecr	iot	macie2	ram	sso
▶ alexaforbusiness	codeguruprofiler	ecr-public	iot-data	managedblockchain	rds	sso-admin
▶ amp	codepipeline	ecs	iot-jobs-data	marketplace-catalog	rds-data	sso-oidc
▶ amplify	codestar	efs	iot1click-devices	marketplace-entitlement	redshift	stepfunctions
▶ amplifybackend	codestar-connections	eks	iot1click-projects	marketplacecommerceanalytics	redshift-data	storagegateway
▶ apigateway	codestar-notifications	elastic-inference	iotanalytics	mediaconnect	rekognition	sts
▶ apigatewaymanagementapi	cognito-identity	elasticache	iotdeviceadvisor	mediaconvert	resource-groups	support
▶ apigatewayv2	cognito-idp	elasticbeanstalk	iotevents	medialive	resourcegroupstaggingapi	swf
▶ appconfig	cognito-sync	elastictranscoder	iotevents-data	mediapackage	robomaker	synthetics
▶ appflow	comprehend	elb	iotfleethub	mediapackage-vod	route53	textract
▶ appintegrations	comprehendmedical	elbv2	iotsecuretunneling	mediastore	route53domains	timestream-query
▶ application-autoscaling	compute-optimizer	emr	iotsitewise	mediastore-data	route53resolver	timestream-write
▶ application-insights	configservice	emr-containers	iotthingsgraph	mediatailor	s3	transcribe
▶ appmesh	configure	es	iotwireless	meteringmarketplace	s3api	transfer
▶ appstream	connect	events	ivs	mgh	s3control	translate
▶ appsync	connect-contact-lens	firehose	kafka	mgn	s3outposts	waf
▶ athena	connectparticipant	fis	kendra	migrationhub-config	sagemaker	waf-regional
▶ auditmanager	cur	fms	kinesis	mobile	sagemaker-a2i-runtime	wafv2
▶ autoscaling	customer-profiles	forecast	kinesis-video-archived-media	mq	sagemaker-edge	wellarchitected
▶ autoscaling-plans	databrew	forecastquery	kinesis-video-media	mturk	sagemaker-featurestore-runtime	workdocs
▶ backup	dataexchange	frauddetector	kinesis-video-signaling	mwaa	sagemaker-runtime	worklink
▶ batch	datapipeline	fsx	kinesisanalytics	neptune	savingsplans	workmail
▶ braket	datasync	gamelift	kinesisanalyticsv2	network-firewall	schemas	workmailmessageflow
▶ budgets	dax	glacier	kinesisvideo	networkmanager	sdb	workspaces
▶ ce	ddb	globalaccelerator	kms	opsworks	secretsmanager	xray
▶ chime	deploy	glue	lakeformation	opsworks-cm	securityhub	
▶ cli-dev	detective	greengrass	lambda	organizations	serverlessrepo	
▶ cloud9	devicefarm	greengrassv2	lex-models	outposts	service-quotas	
▶ clouddirectory	devops-guru	groundstation	lex-runtime	personalize	servicecatalog	
▶ cloudformation	directconnect	guardduty	lexv2-models	personalize-events	servicecatalog-appregistry	
▶ cloudfront	discovery	health	lexv2-runtime	personalize-runtime	servicediscovery	
▶ cloudhsm	dlm	healthlake	license-manager	pi	ses	
▶ cloudhsmv2	dms	help	lightsail	pinpoint	sesv2	
▶ cloudsearch	docdb	history	location	pinpoint-email	shield	
▶ cloudsearchdomain	ds	honeycode	logs	pinpoint-sms-voice	signer	
▶ cloudtrail	dynamodb	iam	lookoutequipment	polly	sms	
▶ cloudwatch	dynamodbstreams	identitystore	lookoutmetrics	pricing	snowball	
▶ codeartifact	ebs	imagebuilder	lookoutvision	qldb	sns	

AWS CLI PREREQUISITES

- ▶ Command Line Basics
- ▶ AWS IAM (**I**dentity & **A**ccess **M**anagement) - Create User
<https://aws.amazon.com/iam/>
- ▶ AWS S3 (**S**imple **S**torage **S**ervice) - Create Bucket, Upload Files
<https://aws.amazon.com/s3/>
- ▶ Download AWS CLI v2 for Linux, Mac, or Windows
<https://aws.amazon.com/cli/>

AWS IAM - CREATE USER

- ▶ Login to AWS Web Console and go to “IAM” Service
<https://console.aws.amazon.com/>
- ▶ Create User Group with policies “AmazonEC2FullAccess” and “AmazonS3FullAccess”, and “AWSCloudFormationFullAccess”
- ▶ Create User with “Programmatic access” and associate with the User Group you just created
- ▶ Download or record the “Access Key ID” and “Secret Access Key”

AWS CLI EXAMPLE - CONFIGURE CREDENTIALS VIA NAMED PROFILE

- ▶ Choose a profile name
- ▶ List existing credentials
- ▶ Configure credentials
- ▶ List new credentials

```
jrk@jrk:~ ok | 11:  
aws configure list --profile demo  
Name          Value        Type        Location  
----          ----        ----        -----  
profile       demo         manual      --profile  
  
The config profile (demo) could not be found  
aws configure --profile demo  
AWS Access Key ID [None]: AKIAVBUATGVZUGQJDPS  
AWS Secret Access Key [None]: k9jthC3IUEzh01VC  
Default region name [None]: us-east-1  
Default output format [None]: json  
aws configure list --profile demo  
Name          Value        Type        Location  
----          ----        ----        -----  
profile       demo         manual      --profile  
access_key    *****JDPS  shared-credentials-file  
secret_key    *****knAP  shared-credentials-file  
region        us-east-1   config-file  ~/.aws/config  
ok | 29s | 11:  
255 | 11: ZgknAP
```

Command Line Extension and Theme

Oh My Zsh <https://ohmyz.sh/>

Powerlevel10k <https://github.com/romkatv/powerlevel10k>

AWS CLI EXAMPLE - USE A NAMED PROFILE

- ▶ Set Profile Name
- ▶ List Credentials

```
jrk@jrk:~ ok | 11:  
~ export AWS_PROFILE=demo  
~ aws configure list  
  
Name Value Type Location  
----  
profile demo manual --profile  
access_key *****JDPS shared-credentials-file  
secret_key *****knAP shared-credentials-file  
region us-east-1 config-file ~/.aws/config  
ok | 11:
```

Command Line Extension and Theme

Oh My Zsh <https://ohmyz.sh/>

Powerlevel 10k <https://github.com/romkatv/powerlevel10k>

AWS CLI EXAMPLE - S3 MANAGEMENT

- ▶ Make Bucket
- ▶ List Bucket Contents
- ▶ Copy File to Bucket
- ▶ Copy File from Bucket
- ▶ Remove File from Bucket
- ▶ Remove Bucket

Command Line Extension and Theme

Oh My Zsh <https://ohmyz.sh/>

Powerlevel10k <https://github.com/romkatv/powerlevel10k>

```
jrk@jrk:~ ok | 6s | demo aws | 11: aws s3 mb s3://sgfaws-cli-sdk
make_bucket: sgfaws-cli-sdk
aws s3 ls s3://sgfaws-cli-sdk
cat hello.txt
ok | 4s | demo aws | 11: ok | 11:
Hello World!
aws s3 cp hello.txt s3://sgfaws-cli-sdk/hello.txt
upload: ./hello.txt to s3://sgfaws-cli-sdk/hello.txt
aws s3 ls s3://sgfaws-cli-sdk
2021-07-07 11:36:47          13 hello.txt
aws s3 cp s3://sgfaws-cli-sdk/hello.txt local.txt
download: s3://sgfaws-cli-sdk/hello.txt to ./local.txt
cat local.txt
ok | demo aws | 11: ok | 11:
Hello World!
aws s3 rm s3://sgfaws-cli-sdk/hello.txt
delete: s3://sgfaws-cli-sdk/hello.txt
aws s3 rb s3://sgfaws-cli-sdk
remove_bucket: sgfaws-cli-sdk
ok | demo aws | 11: ok | 11:
```

AWS CLI EXAMPLE - EC2 MANAGEMENT

- ▶ Describe Instances with JQ
- ▶ Describe Instance Status
- ▶ Start an Instance

```
jrk@jrk:~  
~ aws ec2 describe-instances | jq '.Reservations[].Instances[] | {InstanceId, InstanceType, AvailabilityZone: .Placement.AvailabilityZone}'  
{  
  "InstanceId": "i-02311e94cefcd1d57",  
  "InstanceType": "t2.micro",  
  "AvailabilityZone": "us-east-1c"  
}  
{  
  "InstanceId": "i-0d164cd2818eb7499",  
  "InstanceType": "t4g.nano",  
  "AvailabilityZone": "us-east-1d"  
}  
~ aws ec2 describe-instance-status | jq '.InstanceStatuses[] | {InstanceId: .InstanceState.Name}'  
{  
  "InstanceId": "i-02311e94cefcd1d57",  
  "InstanceState": "running"  
}  
~ aws ec2 start-instances --instance-ids i-0d164cd2818eb7499  
~  
ok | 12:
```

Command Line Extension and Theme

Oh My Zsh <https://ohmyz.sh/>

Powerlevel10k <https://github.com/romkatv/powerlevel10k>

AWS CLI EXAMPLE - CLOUDFORMATION TEMPLATE

- ▶ Create Template
- ▶ Create Stack
- ▶ Describe Stack
- ▶ Delete Stack

Command Line Extension and Theme

Oh My Zsh <https://ohmyz.sh/>

Powerlevel 10k <https://github.com/romkatv/powerlevel10k>

```
vi s3.yaml
AWSTemplateFormatVersion: 2010-09-09
Description: Create Amazon S3 Bucket
Resources:
  S3Bucket:
    Type: AWS::S3::Bucket
    Description: Create an S3 Bucket with this name
    Properties:
      BucketName: sgfaws-demo-cli-cf
      AccessControl: Private
      PublicAccessBlockConfiguration:
        BlockPublicAcls: true
        BlockPublicPolicy: true
        IgnorePublicAcls: true
        RestrictPublicBuckets: true
      BucketEncryption:
        ServerSideEncryptionConfiguration:
          - ServerSideEncryptionByDefault:
              SSEAlgorithm: AES256
      VersioningConfiguration:
        Status: Enabled
Outputs:
  S3Bucket:
```

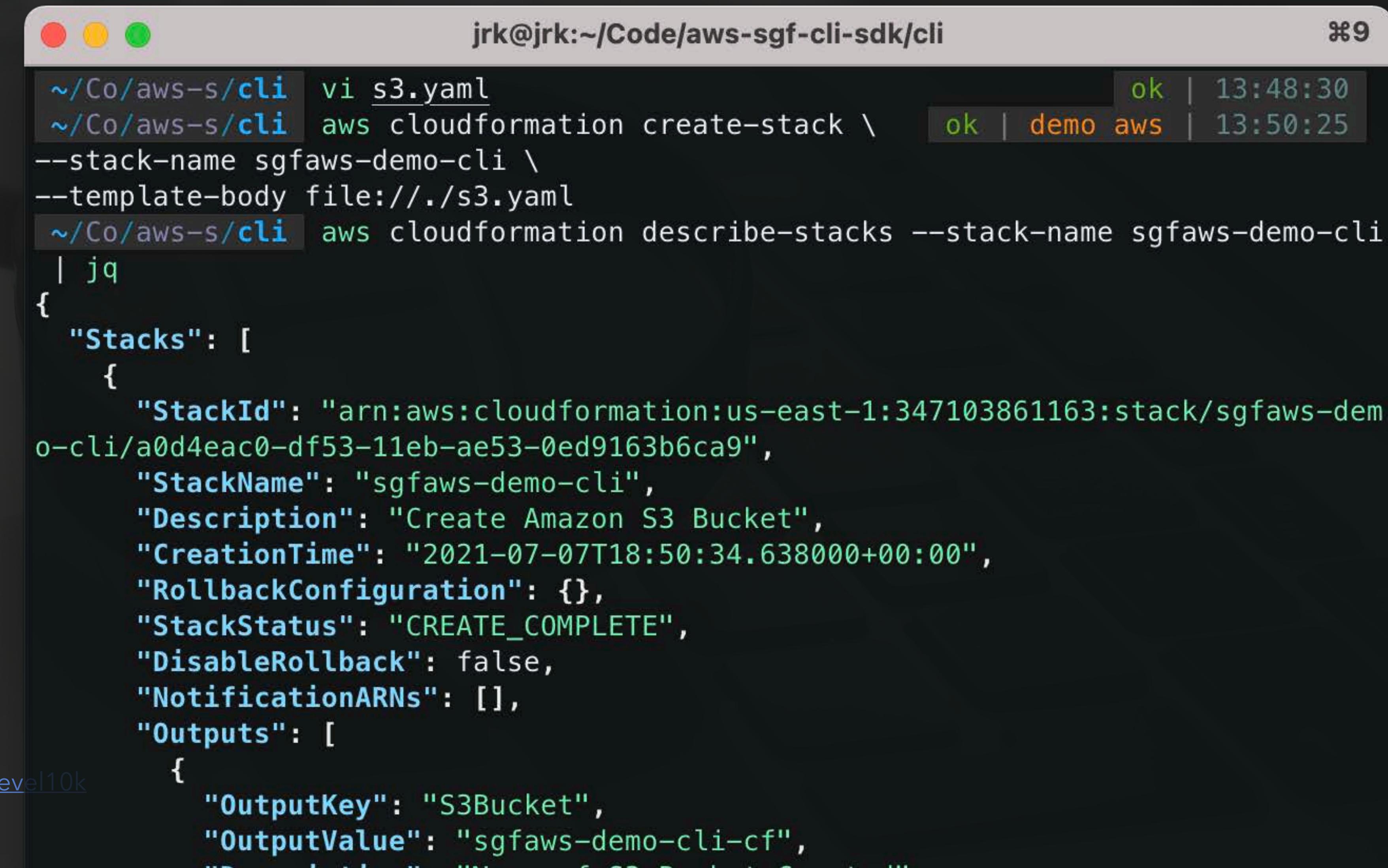
AWS CLI EXAMPLE - CLOUDFORMATION TEMPLATE

- ▶ Create Template
- ▶ Create Stack
- ▶ Describe Stack
- ▶ Delete Stack

Command Line Extension and Theme

Oh My Zsh <https://ohmyz.sh/>

Powerlevel 10k <https://github.com/romkatv/powerlevel10k>



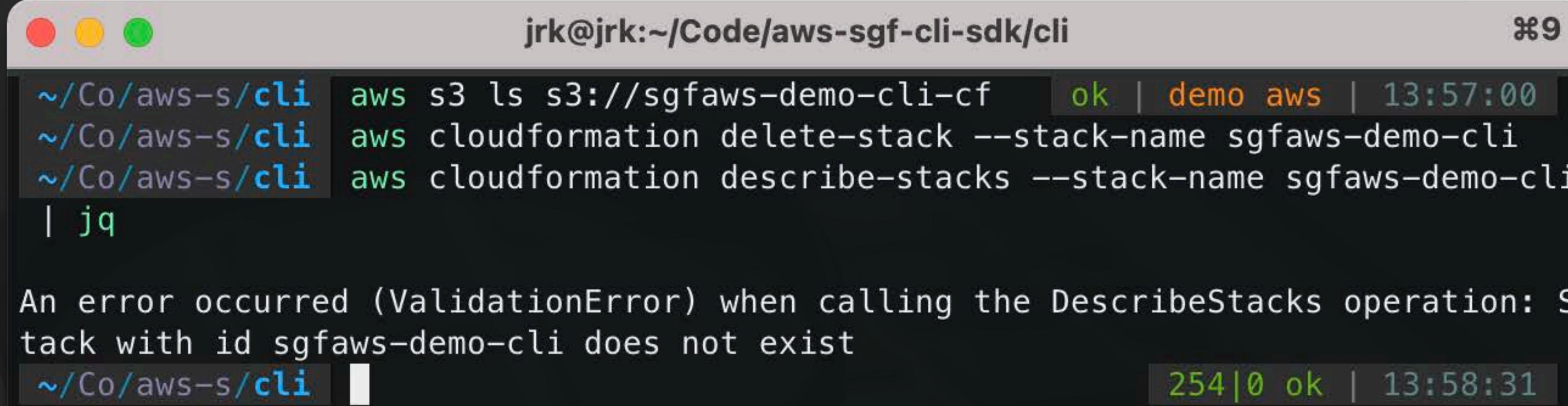
The screenshot shows a terminal window with the following session:

```
jrk@jrk:~/Code/aws-sgf-cli-sdk/cli
~/Co/aws-s/cli vi s3.yaml
~/Co/aws-s/cli aws cloudformation create-stack \
--stack-name sgfaws-demo-cli \
--template-body file://./s3.yaml
~/Co/aws-s/cli aws cloudformation describe-stacks --stack-name sgfaws-demo-cli
| jq
{
  "Stacks": [
    {
      "StackId": "arn:aws:cloudformation:us-east-1:347103861163:stack/sgfaws-de
o-cli/a0d4eac0-df53-11eb-ae53-0ed9163b6ca9",
      "StackName": "sgfaws-demo-cli",
      "Description": "Create Amazon S3 Bucket",
      "CreationTime": "2021-07-07T18:50:34.638000+00:00",
      "RollbackConfiguration": {},
      "StackStatus": "CREATE_COMPLETE",
      "DisableRollback": false,
      "NotificationARNs": [],
      "Outputs": [
        {
          "OutputKey": "S3Bucket",
          "OutputValue": "sgfaws-demo-cli-cf",
          "Description": "The Amazon S3 Bucket created by the CloudFormation stack."}]}
```

The terminal shows the creation of a CloudFormation stack named 'sgfaws-demo-cli' from a local YAML template. It then describes the stack and pipes the JSON output to 'jq' for pretty-printing. The resulting JSON object contains the stack's ID, name, description, creation time, status, and its single output, which is an S3 bucket named 'sgfaws-demo-cli-cf'. The stack status is shown as 'CREATE_COMPLETE'.

AWS CLI EXAMPLE - CLOUDFORMATION TEMPLATE

- ▶ Create Template
- ▶ Create Stack
- ▶ Describe Stack
- ▶ Delete Stack



The screenshot shows a terminal window with the following session:

```
jrk@jrk:~/Code/aws-sgf-cli-sdk/cli
~/Co/aws-s/cli aws s3 ls s3://sgfaws-demo-cli-cf | ok | demo aws | 13:57:00
~/Co/aws-s/cli aws cloudformation delete-stack --stack-name sgfaws-demo-cli
~/Co/aws-s/cli aws cloudformation describe-stacks --stack-name sgfaws-demo-cli
| jq
An error occurred (ValidationError) when calling the DescribeStacks operation: Stack with id sgfaws-demo-cli does not exist
~/Co/aws-s/cli | 254 | 0 ok | 13:58:31
```

The terminal shows a series of AWS CLI commands being run. The first command lists objects in an S3 bucket. The second command attempts to delete a CloudFormation stack named 'sgfaws-demo-cli'. The third command describes stacks in the same region. The fourth command uses the `jq` utility to process the JSON output of the third command. An error message is displayed for the stack deletion attempt, indicating it does not exist. The session ends with a timestamp of 13:58:31.

Command Line Extension and Theme

Oh My Zsh <https://ohmyz.sh/>

Powerlevel 10k <https://github.com/romkatv/powerlevel10k>

AWS CLI RESOURCES

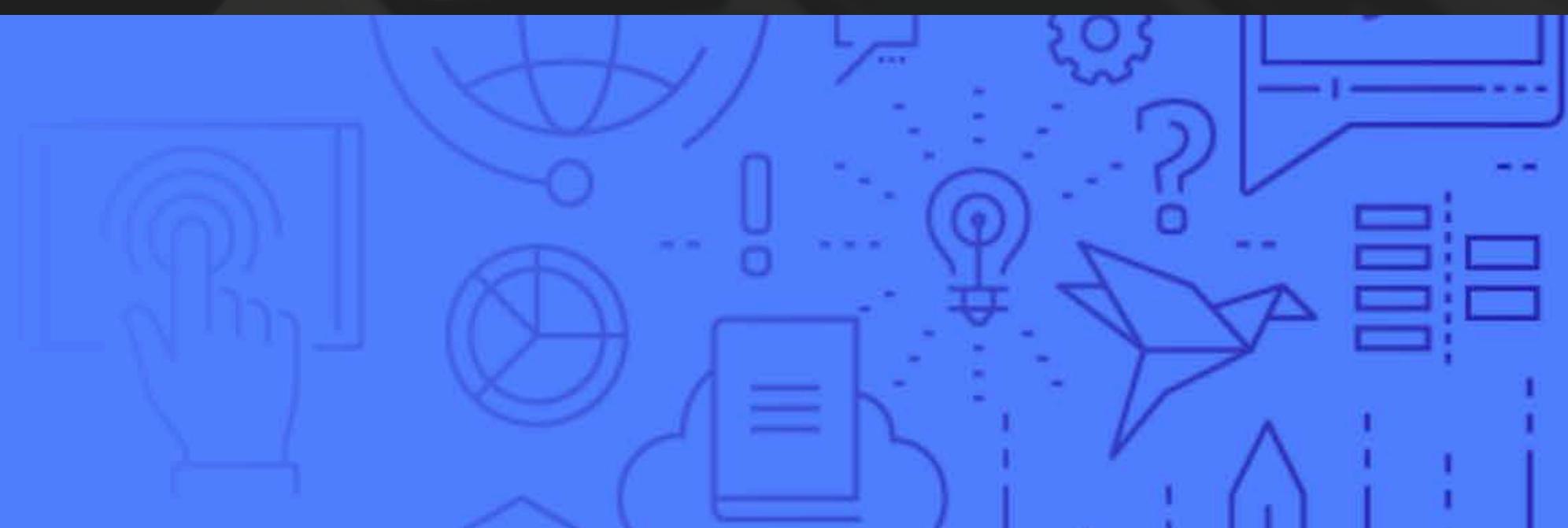
- ▶ AWS Command Line Interface - Official Downloads and Reference
<https://aws.amazon.com/cli/>
- ▶ AWS CLI Named Profile Reference
<https://docs.aws.amazon.com/cli/latest/userguide/cli-configure-profiles.html>
- ▶ JQ (JSON Query) - Command Line JSON Processor
<https://stedolan.github.io/jq/>
- ▶ Amazon S3 Tools: S3cmd Sync - More Powerful Sync Tools for CLI
<https://s3tools.org/s3cmd-sync>
- ▶ Commands and Code Samples
<https://github.com/sgf-aws/sgfaws-intro-cli-sdk>

SDK BENEFITS

- ▶ Control AWS services within your Application
- ▶ Automate infrastructure through your Application
- ▶ Faster and more efficient than AWS web console,
especially for frequent or complex tasks.

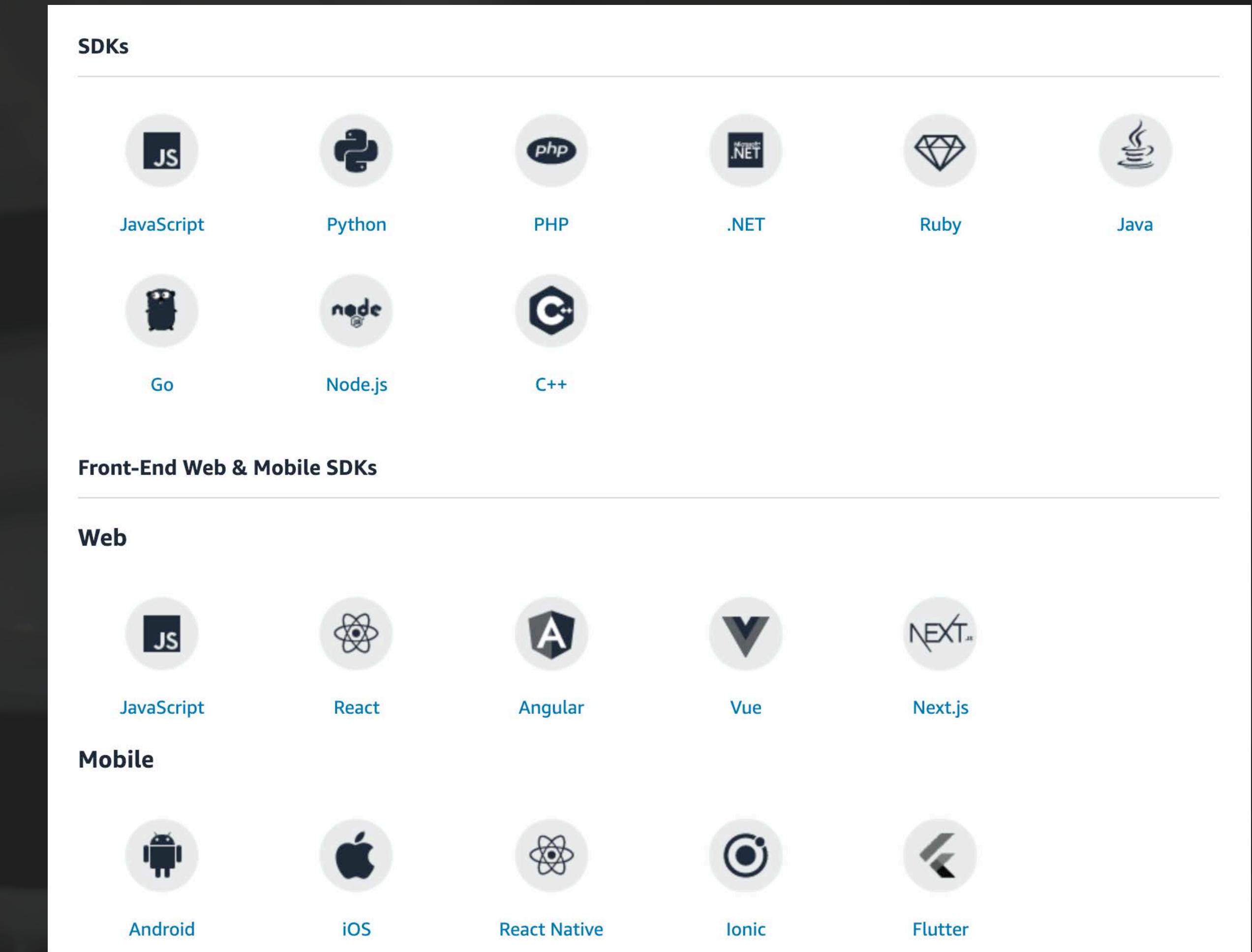
Start building with SDKs and tools

Access and manage AWS services with your preferred development language or platform



SDK FEATURES

- ▶ Access and manage AWS services with your preferred development language or platform
- ▶ SDK converts the AWS REST API into your application language.
- ▶ SDK implements asynchronous requests, retries, timeouts, paginators, etc.



AWS SDK PREREQUISITES

- ▶ Programming Basics
- ▶ AWS IAM (**I**dentity & **A**ccess **M**anagement) - Create User
<https://aws.amazon.com/iam/>
- ▶ AWS S3 (**S**imple **S**torage **S**ervice) - Create Bucket, Upload Files
<https://aws.amazon.com/s3/>
- ▶ Download Version 3 of the AWS SDK for PHP
<https://aws.amazon.com/sdk-for-php/>

AWS IAM - CREATE USER

- ▶ Login to AWS Web Console and go to “IAM” Service
<https://console.aws.amazon.com/>
- ▶ Create User Group with policies “AmazonEC2FullAccess” and “AmazonS3FullAccess”, and “AWSCloudFormationFullAccess”
- ▶ Create User with “Programmatic access” and associate with the User Group you just created
- ▶ Download or record the “Access Key ID” and “Secret Access Key”

AWS SDK EXAMPLE – CONFIGURE CREDENTIALS VIA NAMED PROFILE

- ▶ Choose a profile name
- ▶ List existing credentials
- ▶ Configure credentials
- ▶ List new credentials

```
jrk@jrk:~ ok | 11:  
aws configure list --profile demo  
Name          Value        Type        Location  
----          ----        ----        -----  
profile       demo        manual     --profile  
  
The config profile (demo) could not be found  
aws configure --profile demo  
AWS Access Key ID [None]: AKIAVBUATGVZUGQJDPS  
AWS Secret Access Key [None]: k9jthC3IUEzh01VC  
Default region name [None]: us-east-1  
Default output format [None]: json  
aws configure list --profile demo  
Name          Value        Type        Location  
----          ----        ----        -----  
profile       demo        manual     --profile  
access_key    *****JDPS  shared-credentials-file  
secret_key    *****knAP  shared-credentials-file  
region        us-east-1   config-file  ~/.aws/config  
ok | 29s | 11:  
255 | 11: ZgknAP
```

Command Line Extension and Theme

Oh My Zsh <https://ohmyz.sh/>

Powerlevel10k <https://github.com/romkatv/powerlevel10k>

AWS SDK EXAMPLE – USE A NAMED PROFILE

- ▶ Set Profile Name
- ▶ List Credentials



The screenshot shows a terminal window with the following content:

```
jrk@jrk:~$ export AWS_PROFILE=demo
jrk@jrk:~$ aws configure list
```

Name	Value	Type	Location
profile	demo	manual	--profile
access_key	*****JDPS	shared-credentials-file	
secret_key	*****knAP	shared-credentials-file	
region	us-east-1	config-file	~/.aws/config

Command Line Extension and Theme

Oh My Zsh <https://ohmyz.sh/>

Powerlevel 10k <https://github.com/romkatv/powerlevel10k>

AWS SDK EXAMPLE – S3 MANAGEMENT

- ▶ Install PHP SDK
- ▶ Create PHP Script
- ▶ Run PHP Script
 - ▶ Put Object
 - ▶ Get Object
 - ▶ Output Object Body

```
jrk@jrk:~/Code/aws-sgf-cli-sdk/sdk
~/Co/aws-s/sdk composer require aws/aws-sdk-php
Using version ^3.185 for aws/aws-sdk-php
./composer.json has been updated
Running composer update aws/aws-sdk-php
Loading composer repositories with package information
Updating dependencies
Nothing to modify in lock file
Installing dependencies from lock file (including require-dev)
Nothing to install, update or remove
Generating autoload files
2 packages you are using are looking for funding.
Use the `composer fund` command to find out more!
~/Co/aws-s/sdk vi index.php
~/Co/aws-s/sdk php index.php
Hello World!
~/Co/aws-s/sdk
```

Command Line Extension and Theme

Oh My Zsh <https://ohmyz.sh/>

Powerlevel10k <https://github.com/romkatv/powerlevel10k>

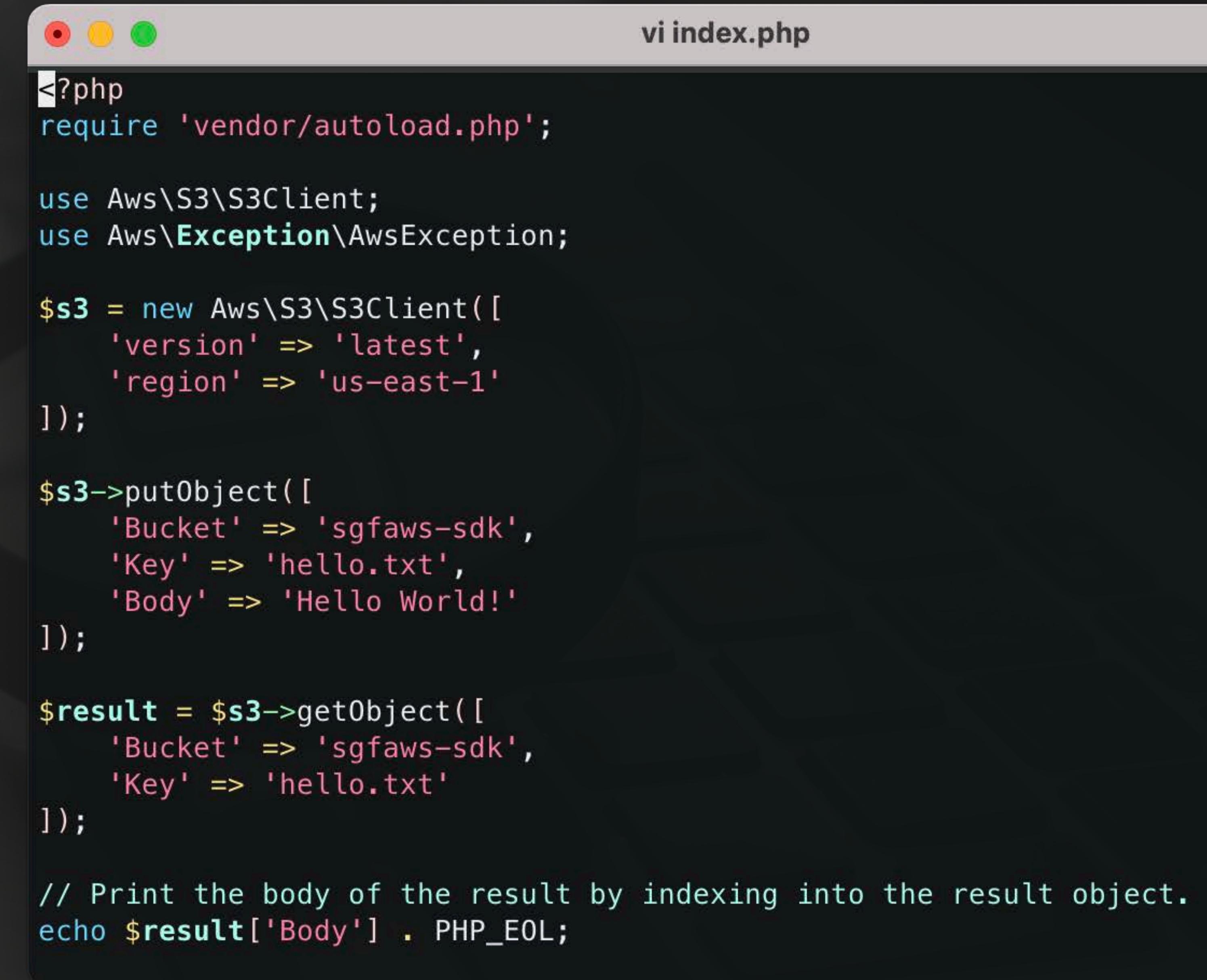
AWS SDK EXAMPLE - S3

- ▶ Install PHP SDK
- ▶ Create PHP Script
- ▶ Run PHP Script
 - ▶ Put Object
 - ▶ Get Object
 - ▶ Output Object Body

Command Line Extension and Theme

Oh My Zsh <https://ohmyz.sh/>

Powerlevel10k <https://github.com/romkatv/powerlevel10k>



```
vi index.php

<?php
require 'vendor/autoload.php';

use Aws\S3\S3Client;
use Aws\Exception\AwsException;

$s3 = new Aws\S3\S3Client([
    'version' => 'latest',
    'region' => 'us-east-1'
]);

$s3->putObject([
    'Bucket' => 'sgfaws-sdk',
    'Key' => 'hello.txt',
    'Body' => 'Hello World!'
]);

$result = $s3->getObject([
    'Bucket' => 'sgfaws-sdk',
    'Key' => 'hello.txt'
]);

// Print the body of the result by indexing into the result object.
echo $result['Body'] . PHP_EOL;
```

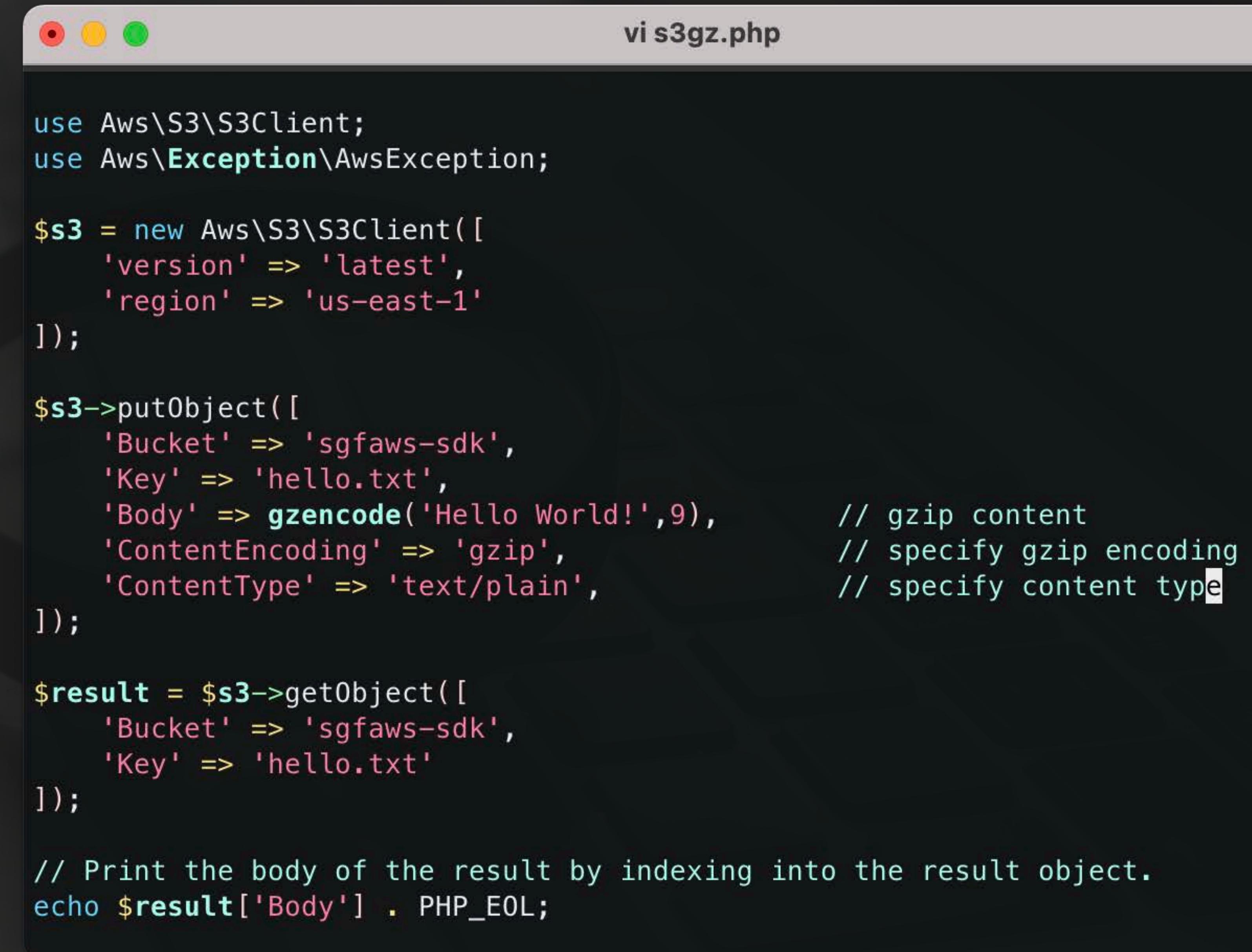
AWS SDK - S3 TIPS

- ▶ Compress content of raw data files with GZIP
- ▶ Specify Content Encoding type of "gzip", then SDK will auto-decompress content!
- ▶ Specify Content Type (e.g. "text/plain" for .txt files, "text/csv" for .csv files)

Command Line Extension and Theme

Oh My Zsh <https://ohmyz.sh/>

Powerlevel10k <https://github.com/romkatv/powerlevel10k>



The screenshot shows a terminal window titled "vi s3gz.php". The code demonstrates how to use the AWS SDK for PHP to upload a compressed file to S3 and then download it back in its original form.

```
use Aws\S3\S3Client;
use Aws\Exception\AwsException;

$s3 = new Aws\S3\S3Client([
    'version' => 'latest',
    'region' => 'us-east-1'
]);

$s3->putObject([
    'Bucket' => 'sgfaws-sdk',
    'Key' => 'hello.txt',
    'Body' => gzencode('Hello World!', 9),           // gzip content
    'ContentEncoding' => 'gzip',                     // specify gzip encoding
    'ContentType' => 'text/plain',                  // specify content type
]);

$result = $s3->getObject([
    'Bucket' => 'sgfaws-sdk',
    'Key' => 'hello.txt'
]);

// Print the body of the result by indexing into the result object.
echo $result['Body'] . PHP_EOL;
```

AWS SDK RESOURCES

- ▶ AWS SDK and Tools - Official Downloads and Reference
<https://aws.amazon.com/getting-started/tools-sdks/>
- ▶ AWS SDK for PHP - Version 3 - Official Reference
<https://docs.aws.amazon.com/sdk-for-php/v3/developer-guide/>
- ▶ Amazon SDK for PHP: putObject() Reference
<https://docs.aws.amazon.com/aws-sdk-php/v3/api/api-s3-2006-03-01.html#putobject>
- ▶ Commands and Code Samples
<https://github.com/sgf-aws/sgfaws-intro-cli-sdk>

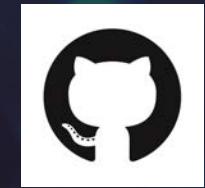
Questions for Jason?

Discuss in our #aws channel in Springfield Devs Discord

► Follow me!



@JasnK



@jason-klein

SPRINGFIELD AMAZON WEB SERVICES USER GROUP
JULY 2021

#SGFAWS

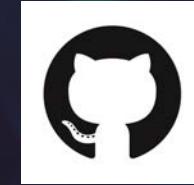
INTRO TO AWS CLI AND SDK: USING COMMAND LINE & DEV TOOLS TO MANAGE AWS

Thank you for attending!

► Follow me!



@JasnK



@jason-klein

SPRINGFIELD AMAZON WEB SERVICES USER GROUP
JULY 2021

#SGFAWS

INTRO TO AWS CLI AND SDK: USING COMMAND LINE & DEV TOOLS TO MANAGE AWS