SSM Document YAML (TKT 59) – Break Glass:

---

schemaVersion: "2.2"

description: "Command to create user using ssm document"

parameters:

UserName:

type: "String"

description: "Create user - break-glass"

default: "break-glass"

mainSteps:

- action: "aws:runShellScript"

name: "CreateUser"

inputs:

runCommand:

- sudo useradd break-glass

- cd /home/break-glass

- sudo mkdir .ssh

- sudo chmod 700 .ssh

- sudo echo -e "ssh-rsa  santiagorivera@MBP-M1.local" >> .ssh/authorized\_keys

- sudo chmod 600 .ssh/authorized\_keys

Splunk:

Warning: splunkforwarder-9.0.3-dd0128b1f8cd-linux-2.6-x86\_64.rpm: Header V4 RSA/SHA256 Signature, key ID b3cd4420: NOKEY

Complete

Checkmk – Ubuntu

Created new site ytmonitoring with version 2.1.0p20.cre.

The site can be started with omd start ytmonitoring.

The default web UI is available at http://ip-172-31-54-49/ytmonitoring/

The admin user for the web applications is cmkadmin with password: YuCwdpbM

For command line administration of the site, log in with 'omd su ytmonitoring'.

After logging in, you can change the password for cmkadmin with 'cmk-passwd cmkadmin'.

oot@ip-172-31-54-49:~# omd start pythonmonitoring

omd: The site 'pythonmonitoring' does not exist. You need to execute omd as root or site user.

root@ip-172-31-54-49:~# omd version

OMD - Open Monitoring Distribution Version 2.1.0p20.cre

root@ip-172-31-54-49:~# omd create pythonmonitoring

Adding /opt/omd/sites/pythonmonitoring/tmp to /etc/fstab.

Creating temporary filesystem /omd/sites/pythonmonitoring/tmp...OK

Updating core configuration...

Generating configuration for core (type nagios)...

Precompiling host checks...OK

Executing post-create script "01\_create-sample-config.py"...OK

Restarting Apache...OK

Created new site pythonmonitoring with version 2.1.0p20.cre.

The site can be started with omd start pythonmonitoring.

The default web UI is available at http://ip-172-31-54-49/pythonmonitoring/

The admin user for the web applications is cmkadmin with password: hiYpfCTp

For command line administration of the site, log in with 'omd su pythonmonitoring'.

After logging in, you can change the password for cmkadmin with 'cmk-passwd cmkadmin'.

RDP:

~~When prompted, connect to your instance using the following details:~~

~~Public DNS~~

~~ec2-18-207-112-79.compute-1.amazonaws.com~~

~~User name~~

~~Administrator~~

~~Password~~

~~p.mU!P4Q%QWjYvJR;)c&0B%\*KEOLL1?@~~

~~If you've joined your instance to a directory, you can use your directory credentials to connect to your instance.~~

When prompted, connect to your instance using the following details:

User name

 Administrator

Password

 p.mU!P4Q%QWjYvJR;)c&0B%\*KEOLL1?@

[Fleet Manager Remote Desktop](https://us-east-1.console.aws.amazon.com/systems-manager/managed-instances/rdp-connect?region=us-east-1&instances=i-09e29235cfe92ed9f)

If you've joined your instance to a directory, you can use your directory credentials to connect to your instance.

New 09Feb23:

root@ip-172-31-60-94:~# omd version

OMD - Open Monitoring Distribution Version 2.1.0p18.cre

root@ip-172-31-60-94:~# omd create ytmonitoring

Adding /opt/omd/sites/ytmonitoring/tmp to /etc/fstab.

Creating temporary filesystem /omd/sites/ytmonitoring/tmp...OK

Updating core configuration...

Generating configuration for core (type nagios)...

Precompiling host checks...OK

Executing post-create script "01\_create-sample-config.py"...OK

Restarting Apache...OK

Created new site ytmonitoring with version 2.1.0p18.cre.

The site can be started with omd start ytmonitoring.

The default web UI is available at http://ip-172-31-60-94/ytmonitoring/

The admin user for the web applications is cmkadmin with password: fHxsHkY9

For command line administration of the site, log in with 'omd su ytmonitoring'.

After logging in, you can change the password for cmkadmin with 'cmk-passwd cmkadmin'.

RDP:

Connection Type

Connect using RDP clientDownload a file to use with your RDP client and retrieve your password.

Connect using Fleet ManagerConnect to your instance using Fleet Manager Remote Desktop.

When prompted, connect to your instance using the following details:

User name

 Administrator

Password

 4d-?Ft@t2NZF\*ta3s9VHlP5Mt\*pMP20k

Brew installation on mac

/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install.sh)"

Chatgpt:

AWS CloudFormation sample EC2, VPC, InternetGateway, Subnets, Routing table, ASG, Launch Template, Target Group, Security Group, User data install Apache, any region, option to select instance type, option to select region, default instance type t2.micro, default region us-east-1a.

TKT-37 Userdata:

#!/bin/bash

sudo yum -y update

sudo yum install -y httpd

sudo systemctl start httpd

sudo systemctl enable httpd

sudo yum install -y git

sudo yum install ruby wget -y

cd /home/ec2-user

sudo wget https://aws-codedeploy-us-east-1.s3.us-east-1.amazonaws.com/latest/install

sudo chmod +x ./install

sudo ./install auto

sudo git config --system credential.helper '!aws codecommit credential-helper $@'

sudo git config --system credential.UseHttpPath true

sudo git config --system credential.UseHttpsPath true

sudo git config --system user.name "Mathewos"

sudo git config --system user.email "matmar2@yahoo.com"

cd /home/ec2-user

sudo git clone -b main <https://git-codecommit.us-east-1.amazonaws.com/v1/repos/TKT-MAT-37-CodeCommit-Repo> /home/ec2-user/TKT-MAT-37-CodeCommit-Repo

vpn site-to-site connection between your own account and the organization account using openswam

<https://www.youtube.com/watch?v=7tTrN8WXMlg&ab_channel=DigitalCloudTraining>

**Lambda needs a role that will allow it to stop and start EC2 instances in your account. Make sure your Role does not allow Lambda to perform other activities.**

Created policy “Start-Stop-Instance-Policy-TKT-62”

Created role “Start-Stop-Instance-Role-TKT-62”

**Policy:**

{  
"Version": "2012-10-17",  
"Statement": [  
{  
"Sid": "VisualEditor0",  
"Effect": "Allow",  
"Action": [  
"ec2:StartInstances",  
"ec2:StopInstances",  
"logs:PutLogEvents"  
],  
"Resource": [  
"arn:aws:license-manager::202618001640:license-configuration:",  
"arn:aws:ec2::202618001640:instance/",  
"arn:aws:logs::202618001640:log-group::log-stream:"  
]  
},  
{  
"Sid": "VisualEditor1",  
"Effect": "Allow",  
"Action": [  
"logs:CreateLogStream",  
"logs:CreateLogGroup"  
],  
"Resource": "arn:aws:logs::202618001640:log-group:\*"  
}  
]  
}

**Role thrust relationship/entities:**

{  
"Version": "2012-10-17",  
"Statement": [  
{  
"Effect": "Allow",  
"Principal": {  
"Service": "[lambda.amazonaws.com](http://lambda.amazonaws.com)"  
},  
"Action": "sts:AssumeRole"  
}  
]  
}

MAT-59 Requirements:

* Create an SSM document to deploy this user so that it can be used again in the future. In your SSM document include some commands that verify that the user has been added.
* Since the user will not be able to login with a password you will need to add the following public Key to the authorized key file.
* ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAACAQDg4Tss2eHWT2Z/6SYYMNb2hzmcl0hoxckGhTf3KVIVpRFrfRks3Qnh2oGycey2icwYfrun1vWm6rmUug9hZTVTHfsabj++xylEu/8XuQto0HeWp9tk4iwNku6EZ+JBzuR7iHrVZcwqSCQn5ln9SaKALO
* uCqoDNQq/gWIYPmZVE0WD+66pIfKuUvaADNpVecn4B+AGsfBgAeKVz63zXqxbR8Y3Hjl2rvVfGBSGj9kxykH9klI0ew3falGh3D5JNpxXRyWg2u9LkLXEqqXRJXF9JqRKT0ZqTLaprWqOe6U7DoHV8ktyYyE6F1WqwjU4g0f8+gYQ8DgQK8ijcwnZmnfAo
* 21RniftymQRoGx6rr3o0TshA61WwC/lI90jnLakt89KWlEvMtb4jWkL0a9NuaUQf21c29lYvQfLSRyqqMXanGrkcrhCG0iY39gwAYVPMotulycd4BEOxv9sVDCoeHcDkJZw4XYVpSJvI1+vh41gquJJLEuA3Sq997uxSrOSjCPncOqSeOd63TJYXYk1hR6
* rbkipoqwglDrRbWKFZo+aWYpBISDV/ap0a9S4c3ltRrwJN6BgI5ZvHWEw0wUQA/VKnRlFtnvNYylIlMFnDjzDhgRxpunvmPpaz1cl1uiM398mNuOyQikxxqZovdkEYveJWssOM/V5ulY9HYGTaRAEO1w== santiagorivera@MBP-M1.local

Here is a link with information on how to accomplish this:

<https://aws.amazon.com/premiumsupport/knowledge-center/new-user-accounts-linux-instance/>

SSM Document:

---

schemaVersion: "2.2"

description: "Command to create user using ssm document"

parameters:

UserName:

type: "String"

description: "Create user - break-glass"

default: "break-glass"

mainSteps:

- action: "aws:runShellScript"

name: "CreateUser"

inputs:

runCommand:

- sudo yum update -y

- sudo useradd break-glass

- sudo su - break-glass

- cd /home/break-glass

- mkdir .ssh

- chmod 700 .ssh

- echo -e "ssh-rsa  santiagorivera@MBP-M1.local" >> .ssh/authorized\_keys

- chmod 600 .ssh/authorized\_keys

# How to install Splunk on a Linux Instance in AWS

<https://www.youtube.com/watch?v=0CHaDfNI4Sg&ab_channel=EmekaakaOn1Productions>

Pritunl Client:

<https://client.pritunl.com/#install>

# AWS EC2 + Autoscaling + Load Balancer + CodeDeploy | Deploy Code At Scale | DevOps With AWS

<https://www.youtube.com/watch?v=Ekgi2HfnJcw&t=1636s&ab_channel=SandipDas>

Github:

HTTPS:

<https://github.com/matmar2/python.git>

Github CLI:

gh repo clone matmar2/python

ChatGPT

<https://openai.com/blog/chatgpt/>

# Setup an AWS Site-to-Site Virtual Private Network (VPN)

<https://www.youtube.com/watch?v=7tTrN8WXMlg>

# AWS VPC Peering Connection Concept with Demo | VPC Peering | AWS VPC Peering Step by Step | AWS Demo

<https://www.youtube.com/watch?v=q-NTKPb16SM>

Terraform for beginners:

<https://kodekloud.com/courses/lab-terraform-for-beginners/>

# ChatGPT Tutorial - Use ChatGPT for DevOps tasks to 10x Your Productivity

<https://www.youtube.com/watch?v=l-kE11fhfaQ>

Checkmk:

<https://checkmk.com/product/features>

# Episode 1: Installing Checkmk and monitoring your first host

<https://www.youtube.com/watch?v=opO-SOgOJ1I&ab_channel=Checkmk>

# How to monitor your Linux servers with Checkmk

<https://www.techrepublic.com/article/how-to-monitor-your-linux-servers-with-checkmk/>

Apprenticeship Site – Procore

<https://sites.google.com/procoreplus.com/apprenticeship-site-aws/tool-guide?authuser=0>

# [ AWS 14 ] Set up your first AWS CodeCommit Repository

<https://www.youtube.com/watch?v=E1GJqfIEJkM&ab_channel=JustmeandOpensource>

### openSUSE, SUSE

<https://packages.cisofy.com/community/#centos-rhel>

# (SAA-C02) AWS Certified Solutions Architect - Associate

<https://www.youtube.com/watch?v=6S0_rNYzcPs&ab_channel=ChienDuong>

Cloudformation and Terraform templates from existing AWS setup - Former2

[www.former2.com](http://www.former2.com)

# Terraform S3 Backend Best Practices

<https://technology.doximity.com/articles/terraform-s3-backend-best-practices>