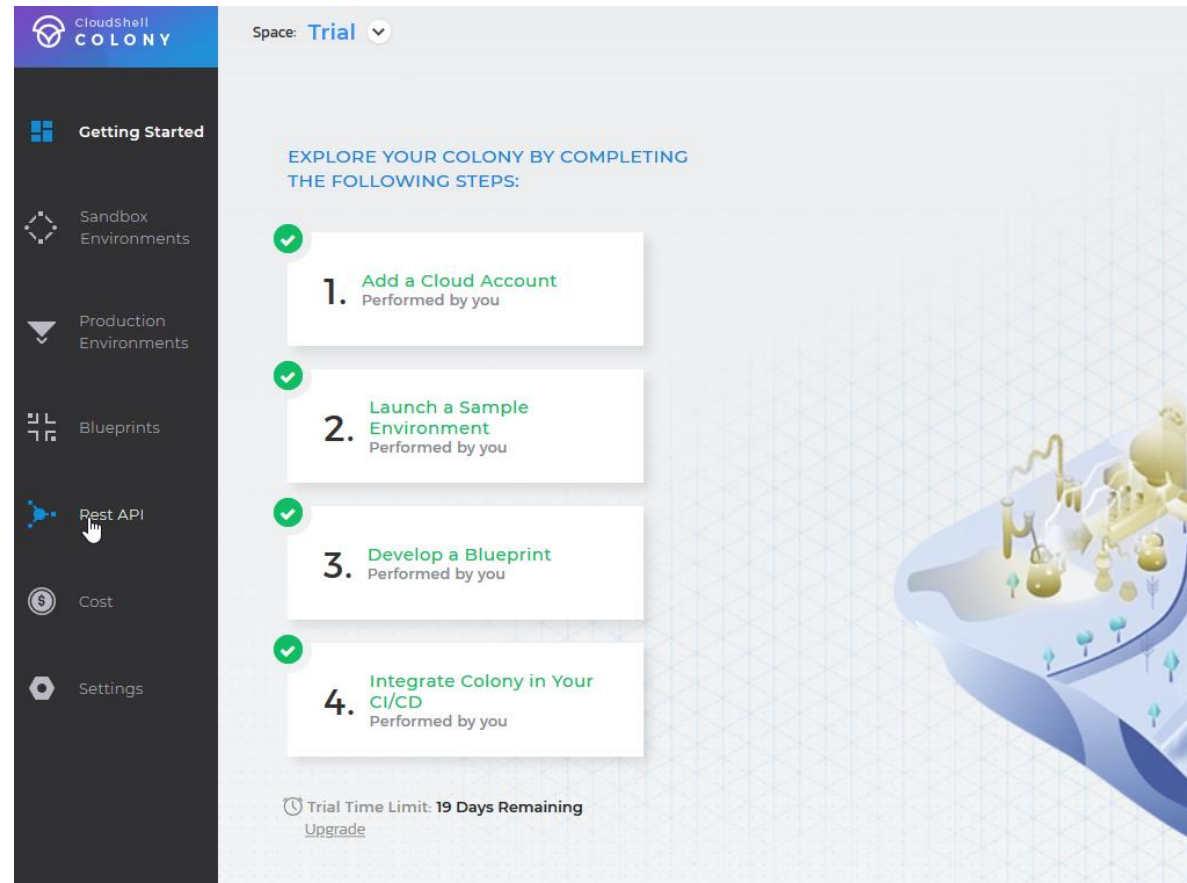


# Module 3


- You are a developer and will use Jenkins to deploy the promotion application environment from the blueprint catalog using the Colony REST API.
- You will run a Jenkins job that will automatically deploy the promotion app, retrieve the URL of the deployed app, run your test and terminate the environment.

# Browse the Colony REST API (right side menu)



This workshop module will call the create SandBox , Get Sandbox details and Terminate Sandbox API methods

**Colony**  
/swagger/latest/swagger.json

Authorize 

Filter by tag

Access Tokens

>

Blueprints

>

Blueprints Catalog

>

Production Environments


>

Sandbox Environments

▼


GET

/api/spaces/{space\_name}/sandbox/{sandboxId}




DELETE

/api/spaces/{space\_name}/sandbox/{sandboxId}



PUT

/api/spaces/{space\_name}/sandbox/{sandboxId}/scheduled\_end\_time



# Deploy and Configure Jenkins

- The first step will be for you to automatically deploy and configure Jenkins using CloudShell Colony.
- This step would typically be performed by a platform Ops/Tools team ahead of time.

# Fork the repository with the Jenkins Blueprint

- <https://github.com/QualiSystemsLab/jenkins-colony>

The screenshot shows the GitHub repository page for `QualiSystemsLab/jenkins-colony`. The repository is owned by `QualiSystemsLab` and has 7 watches, 0 stars, and 1 fork. The repository is a blueprint for Jenkins deployment, containing files like `applications`, `blueprints`, `LICENSE`, and `README.md`. The `README.md` file is visible, showing the title "CloudShell Colony blueprint repository for Jenkins deployment".

Repository details:

- Repository: `QualiSystemsLab/jenkins-colony`
- Watches: 7
- Stars: 0
- Forks: 1

Repository contents:

File/Folder	Description	Last Commit
<code>applications</code>	Support both us-west-1 and us-east-1 in apps	3 days ago
<code>blueprints</code>	change PM region	2 days ago
<code>LICENSE</code>	Initial implementation w/0 artifacts	28 days ago
<code>README.md</code>	Add initial content for Reeadme	23 hours ago

Repository description:

CloudShell Colony blueprint repository for Jenkins deployment

# Create a new “Ops-Tools” space in Colony

The screenshot displays the CloudShell COLONY user interface. On the left is a dark sidebar with navigation links: 'Back to My Space', 'Cloud Accounts', 'Spaces' (highlighted), 'Users', 'Roles', and 'Account & Billing'. The main content area is titled 'Spaces' and includes a search bar. Below the search bar is a table with the following data:

SPACE NAME	SPACE ADMINS	USERS	CLOUD ACCOUNTS	
Trial	pascal joly	1	AWS	<a href="#">Manage &gt;</a> ...

In the foreground, a 'Create a New Space' modal is open. It has a 'GENERAL' tab and a 'Space Name' input field containing the text 'Ops-Tools'. At the bottom of the modal are two buttons: 'Cancel' and 'Next'.

# You will use this space to manage Ops platforms such as Jenkins

CloudShell COLONY

< Back to My Space

Cloud Accounts

**Spaces**

Users






Roles

Account & Billing

Spaces

Search by name

+ Create Space

SPACE NAME	SPACE ADMINS	USERS	CLOUD ACCOUNTS
Ops-Tools	 pascal joly	1	<a href="#">Manage &gt;</a> 
Trial	 pascal joly	1	 AWS <a href="#">Manage &gt;</a> 

# Add your AWS cloud account to the Ops-Tools space

[← Back to My Space](#)

Cloud Accounts

Spaces

Users

Roles

Account & Billing

Cloud Accounts

ACCOUNT


CLOUD PROVIDER ID

AUTHENTICATED

AUTHENTICATED BY

USED IN SPACE

COMPUTE SERVICES

 AWS


754261393455

Oct 1, 2020 11:14 PM

PJ

 pascal joly

Trial



...

Edit

Validate

Delete



Edit Cloud Account

GENERAL

AWS

SPACES

Select space

Add

^ Trial

☒ Internet facing ⓘ

☐ Use shared Virtual Network ⓘ

Cancel

Save

Edit Cloud Account

GENERAL

AWS

SPACES

Select space

Add

^ Trial

^ Ops-Tools

☒ Internet facing ⓘ

☐ Use shared Virtual Network ⓘ

Cancel

Save

Cloud Accounts

+ Add Cloud Account

Search by name

ACCOUNT

CLOUD PROVIDER ID

AUTHENTICATED

AUTHENTICATED BY

USED IN SPACE

COMPUTE SERVICES



754261393455

Oct 1, 2020 11:14 PM



pascal joly

Trial, Ops-Tools



# Add blueprint repository to the Ops-Tools space

CloudShell COLONY

Space: **Ops-Tools** ▼

General **Blueprint Repositories** Cloud Accounts Users Integrations

◆ BLUEPRINT REPOSITORY FOR SANDBOX ENVIRONMENTS

Start by adding a repository for Sandbox Environments

+ Add a Repository

▼ BLUEPRINT REPOSITORY FOR PRODUCTION ENVIRONMENTS

+ Add a Repository

Getting Started

Sandbox Environments

Production Environments

Blueprints

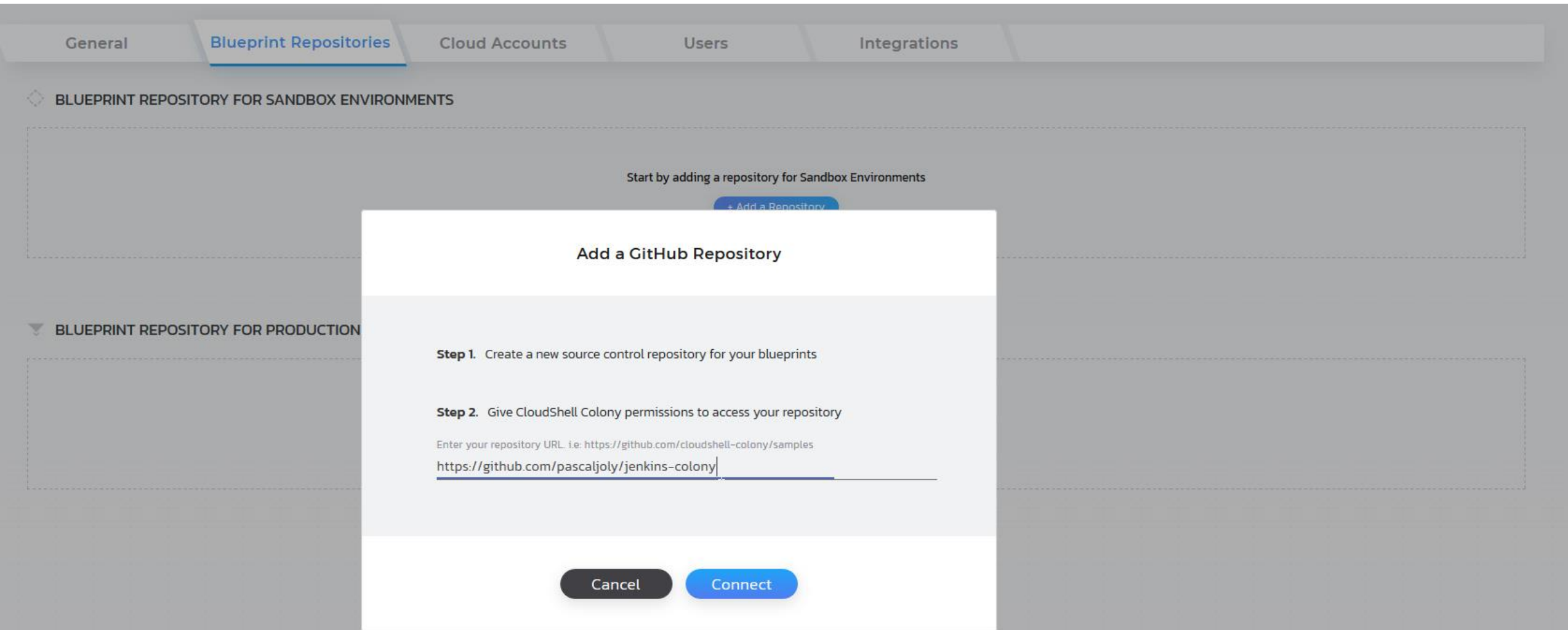
Rest API

Cost

Settings

pascal joly ▼

# Use the Jenkins repo you forked earlier in Git



# The Jenkins blueprint is now listed in your blueprint catalog

General

Blueprint Repositories

Cloud Accounts

Users

Integrations

BLUEPRINT REPOSITORY FOR SANDBOX ENVIRONMENTS

https://github.com/pascaljoly/jenkins-colony

Disconnect

NEXT: DEVELOP YOUR FIRST BLUEPRINT

[Check out our tutorials](#)

Space: Ops-Tools

🔊 📍 👤 pascal joly

Your space's blueprints are source-controlled. [view on Github](#). To learn more about creating blueprints, see [this Help Center section](#).

Space Blueprints

Sample Blueprints

Toggle to publish a blueprint to the self-service catalog.

PUBLISHED	BLUEPRINTS NAME	APPLICATIONS	CLOUD	LAST MODIFIED	MODIFIED BY
<div></div>	Jenkins ⓘ	<div>1 Apps</div>	<div>AWS US West (N. California)</div>	Oct 9, 2020 3:43 PM	<div>outdoorclone</div>

# Go to Settings and add the artifact repository (S3 bucket)

The screenshot displays the CloudShell COLONY user interface. On the left is a dark sidebar with navigation links: Getting Started, Sandbox Environments, Production Environments, Blueprints, Rest API, Cost, and Settings (highlighted with a blue gear icon). The main content area has a top bar showing 'Space: Ops-Tools' and a user profile 'pascal joly'. Below this is a tabbed interface with 'Integrations' selected. The 'Integrations' tab is divided into two sections: 'CI AND PIPELINE TOOLS' and 'ARTIFACT REPOSITORIES'. The 'CI AND PIPELINE TOOLS' section lists Jenkins, Teamcity, Azure DevOps, Bamboo, and Circle CI, each with a 'Connect' button. The 'ARTIFACT REPOSITORIES' section is currently empty, featuring a dashed border and a '+ Add an Artifact Repository' button.

CloudShell COLONY

Space: Ops-Tools

pascal joly

General Blueprint Repositories Cloud Accounts Users Integrations

CI AND PIPELINE TOOLS

Jenkins Teamcity Azure DevOps

Connect Connect Connect

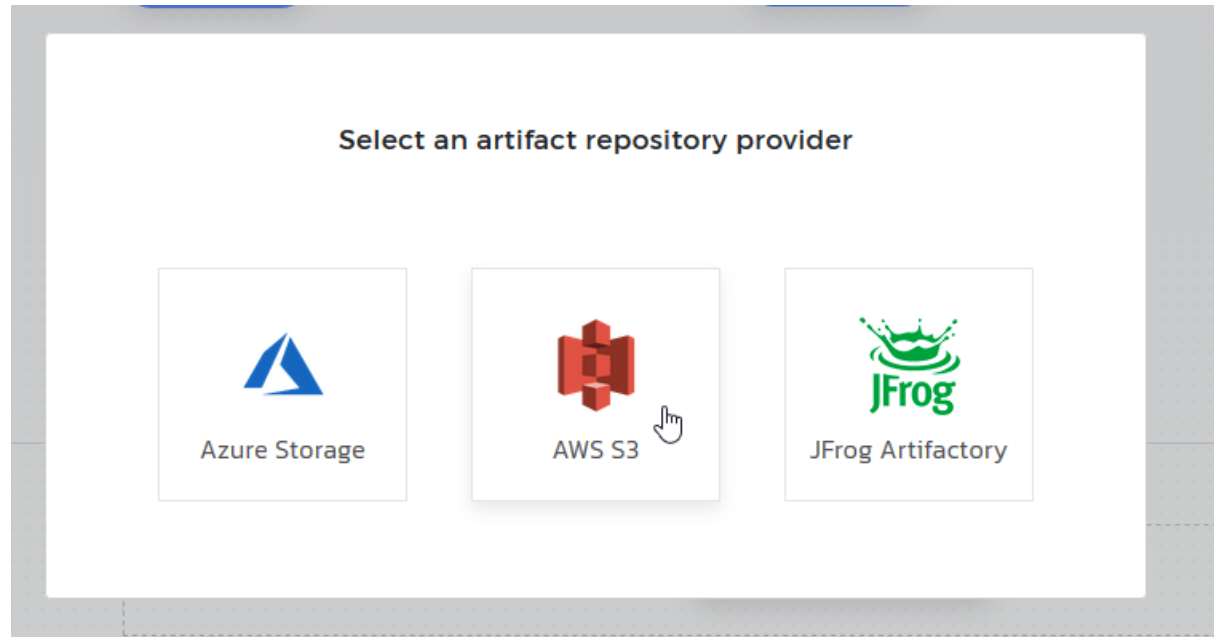
Bamboo Circle CI

Connect Connect


ARTIFACT REPOSITORIES

+ Add an Artifact Repository


# Select AWS S3




# Select from the pull down the cloud provider (AWS) and the S3 repo you previously created in Module 1

Add an artifact repository 

AWS Account

AWS 

Bucket Name

artifact-repo-promotionapp 

Cancel

Confirm

## ARTIFACT REPOSITORIES



s3

Last Update: 10/12/2020, 7:53:50 PM

Disconnect

Bucket: artifact-repo-promotionapp

# Publish Jenkins Blueprint: toggle the slider


Space: **Ops-Tools** ▼









Your space's blueprints are source-controlled. [view on Github](#). To learn more about creating blueprints, see [this Help Center section](#).

✓ You published your first blueprint. A giant leap for all space members!

Space Blueprints

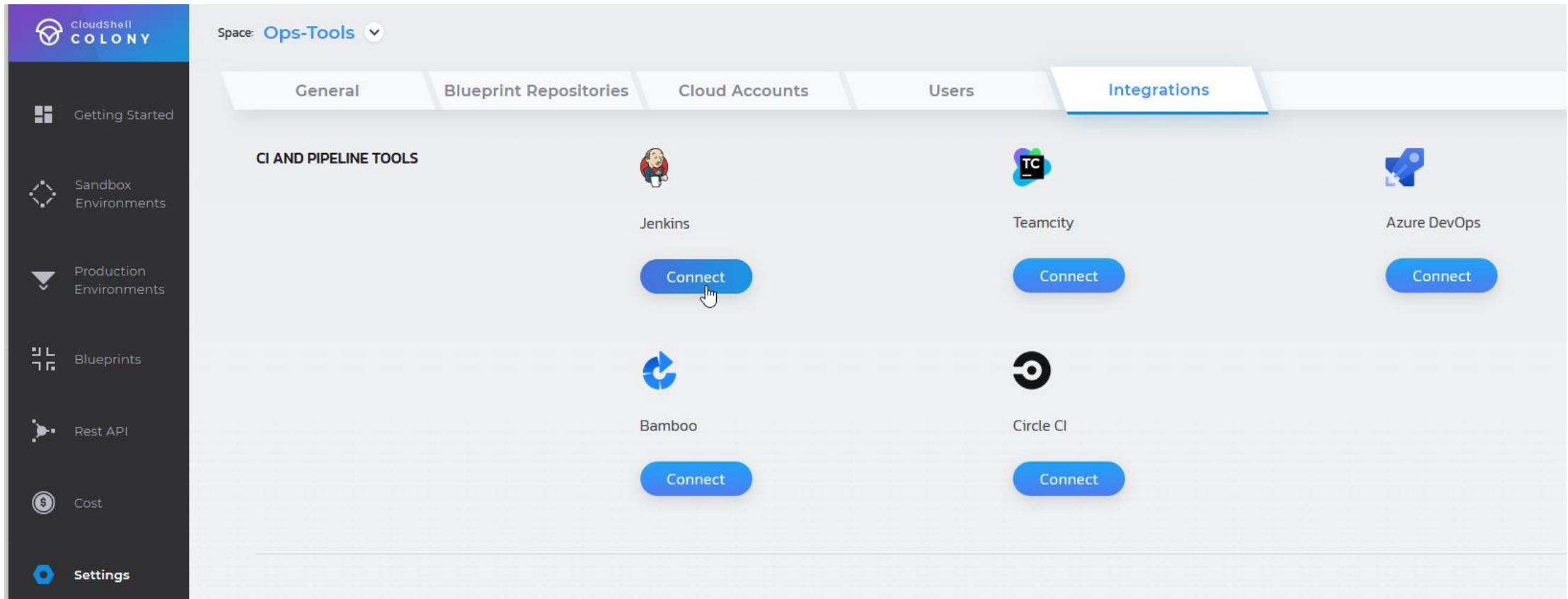
Sample Blueprints

 Toggle to publish a blueprint to the self-service catalog.

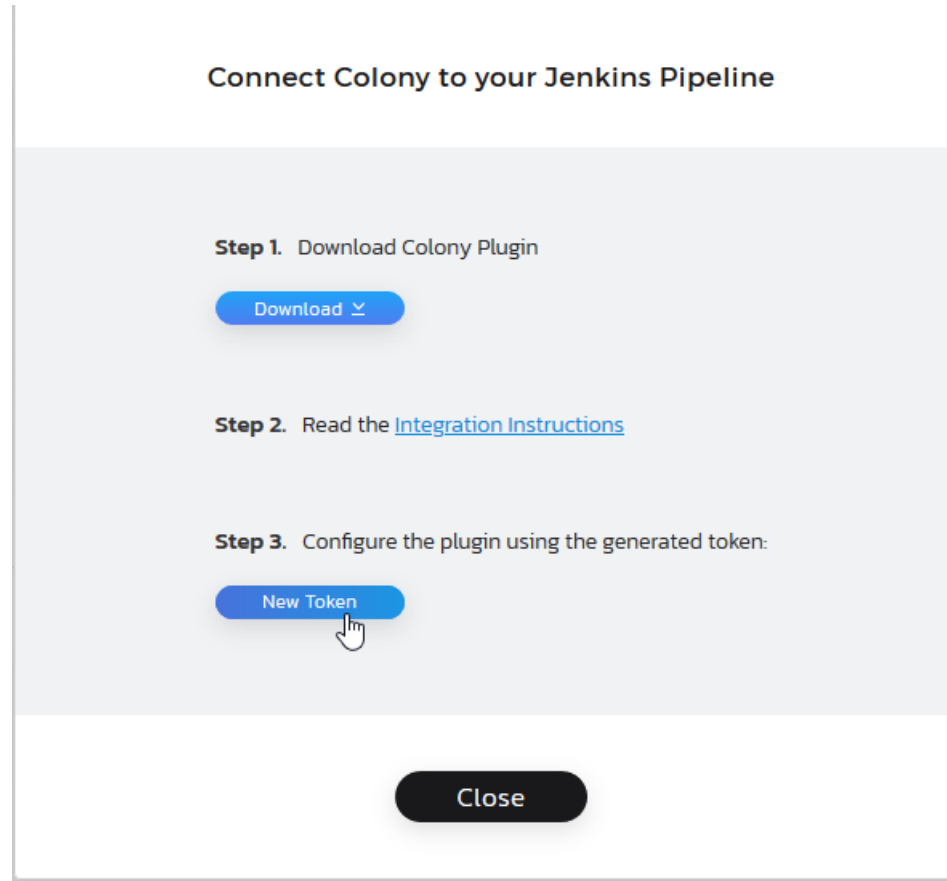
PUBLISHED	BLUEPRINTS NAME	APPLICATIONS	CLOUD	LAST MODIFIED	MODIFIED BY
	Jenkins ⓘ	 1 Apps	 AWS US West (N. California)	Oct 9, 2020 3:43 PM	 outdoorclone
	Promotions Manager AWS ⓘ	 3 Apps	 AWS US West (N. California)	Oct 9, 2020 3:43 PM	 outdoorclone



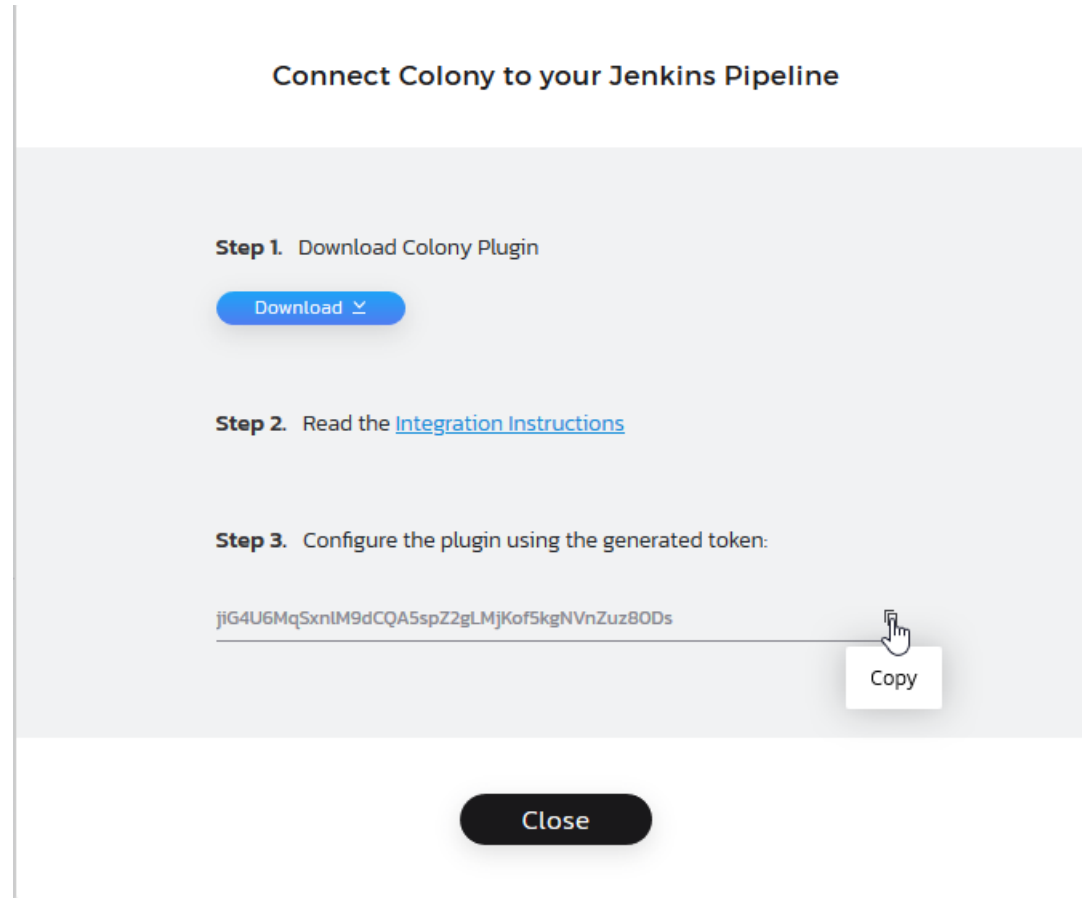
# Navigate to Settings > Integrations and click Connect under the Jenkins icon



# Generate Colony token for Jenkins: click on New Token

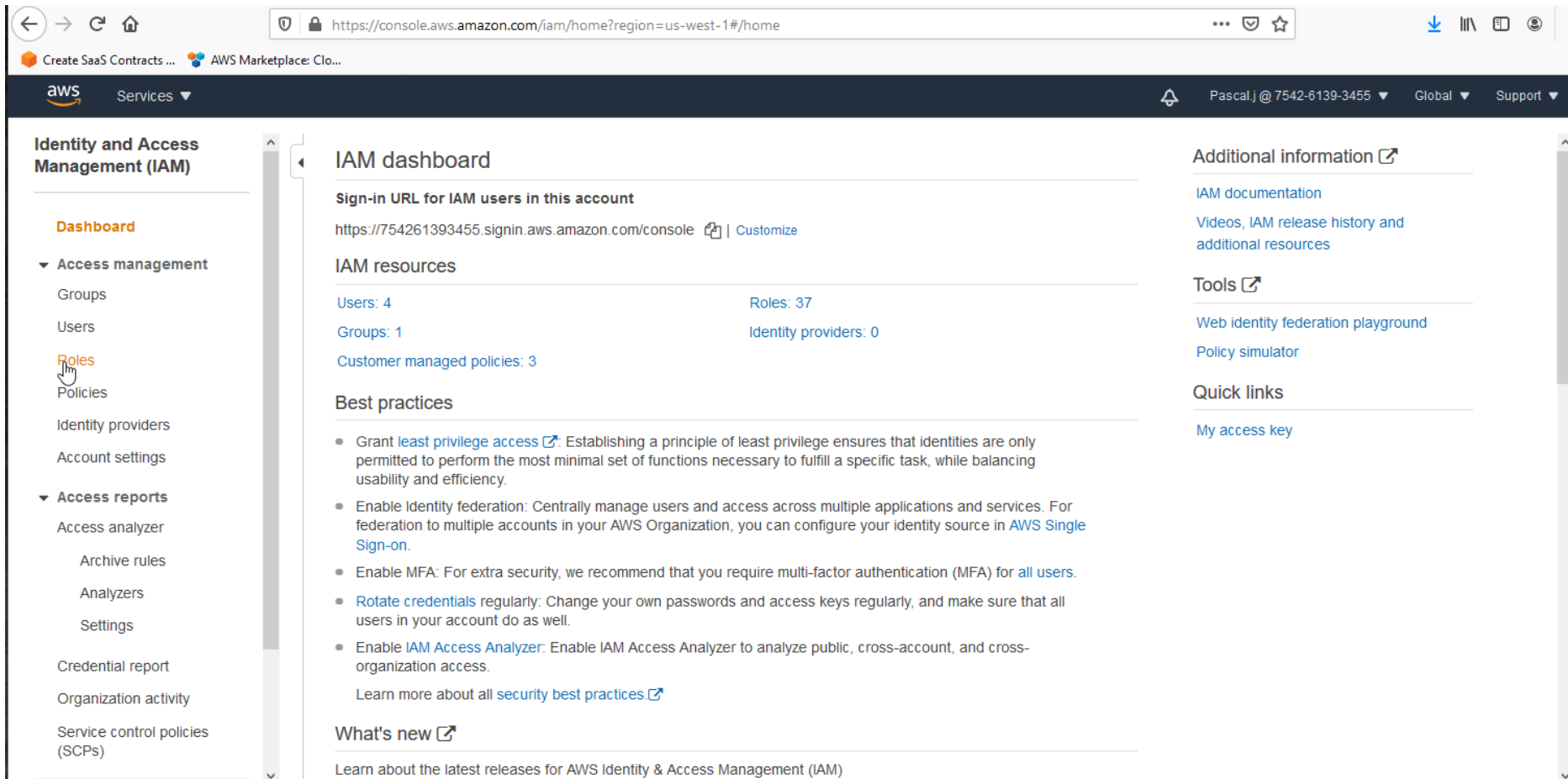


Save the token in a notepad for future use: this will be an input parameter of your sandbox



# In AWS Console, create a IAM role with write access to the S3 bucket

## Navigate to Services > IAM



The screenshot shows the AWS IAM console dashboard. The left sidebar contains the navigation menu with 'Roles' highlighted. The main content area displays the 'IAM dashboard' with a sign-in URL, IAM resource counts (Users: 4, Groups: 1, Roles: 37, Identity providers: 0, Customer managed policies: 3), best practices, and a 'What's new' section. The right sidebar contains additional information, tools, and quick links.

**Identity and Access Management (IAM)**

- Dashboard
- Access management
  - Groups
  - Users
  - Roles**
  - Policies
  - Identity providers
  - Account settings
- Access reports
  - Access analyzer
    - Archive rules
    - Analyzers
    - Settings
  - Credential report
  - Organization activity
  - Service control policies (SCPs)

**IAM dashboard**

Sign-in URL for IAM users in this account  
<https://754261393455.signin.aws.amazon.com/console> | [Customize](#)

**IAM resources**

Users: 4	Roles: 37
Groups: 1	Identity providers: 0
Customer managed policies: 3	

**Best practices**

- Grant [least privilege access](#): Establishing a principle of least privilege ensures that identities are only permitted to perform the most minimal set of functions necessary to fulfill a specific task, while balancing usability and efficiency.
- Enable identity federation: Centrally manage users and access across multiple applications and services. For federation to multiple accounts in your AWS Organization, you can configure your identity source in [AWS Single Sign-on](#).
- Enable MFA: For extra security, we recommend that you require multi-factor authentication (MFA) for [all users](#).
- [Rotate credentials](#) regularly: Change your own passwords and access keys regularly, and make sure that all users in your account do as well.
- Enable [IAM Access Analyzer](#): Enable IAM Access Analyzer to analyze public, cross-account, and cross-organization access.  
[Learn more about all security best practices](#)

**What's new**

[Learn about the latest releases for AWS Identity & Access Management \(IAM\)](#)

**Additional information**

- [IAM documentation](#)
- [Videos, IAM release history and additional resources](#)

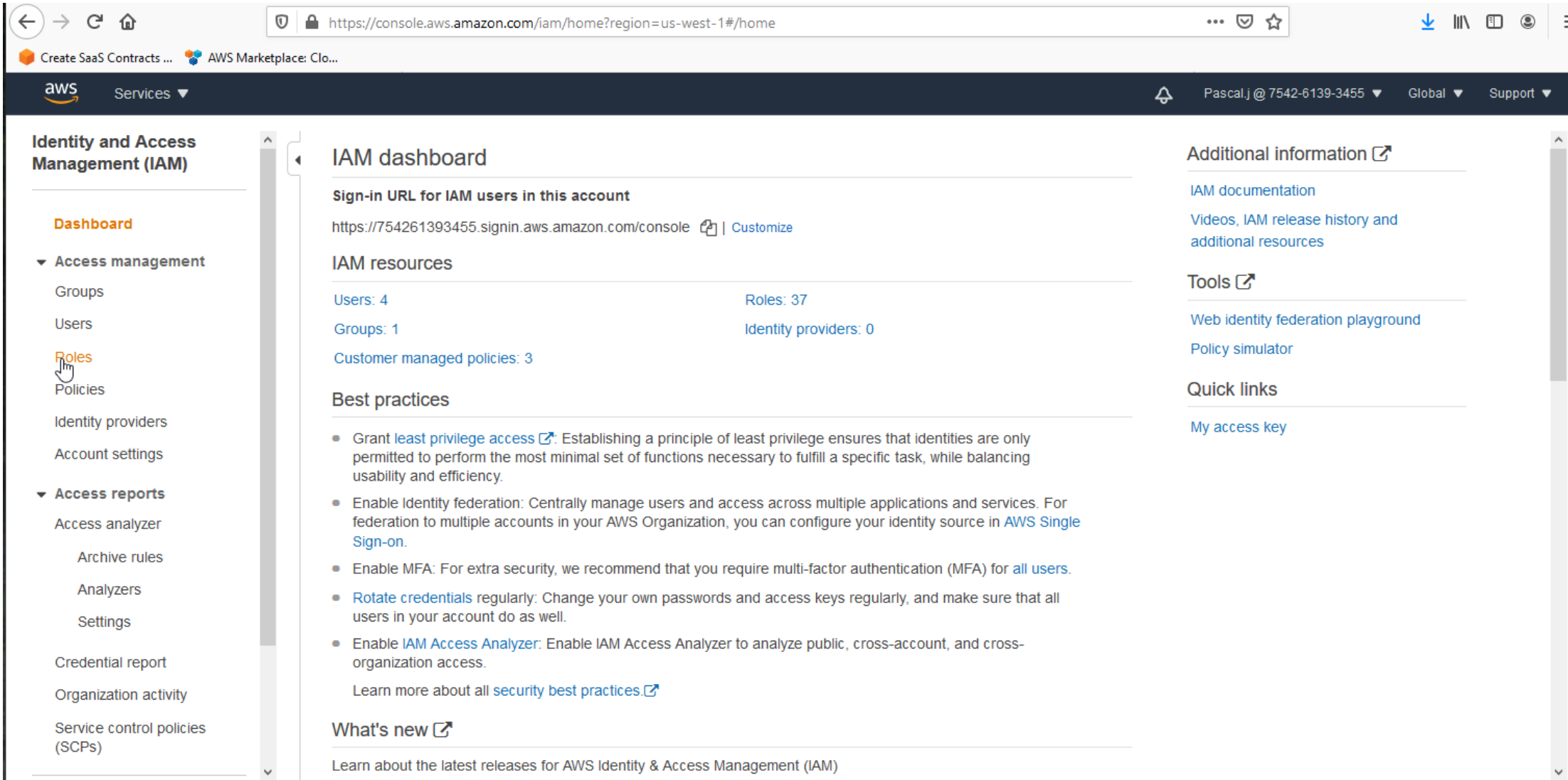
**Tools**

- [Web identity federation playground](#)
- [Policy simulator](#)

**Quick links**

- [My access key](#)

# Navigate to the IAM service dashboard and click on Roles



The screenshot shows the AWS IAM console dashboard. The left sidebar is titled 'Identity and Access Management (IAM)' and contains a 'Dashboard' section with a list of links: 'Groups', 'Users', 'Roles' (highlighted with a hand cursor), 'Policies', 'Identity providers', 'Account settings', 'Access reports', 'Access analyzer', 'Archive rules', 'Analyzers', 'Settings', 'Credential report', 'Organization activity', and 'Service control policies (SCPs)'. The main content area is titled 'IAM dashboard' and displays the 'Sign-in URL for IAM users in this account' as 'https://754261393455.signin.aws.amazon.com/console'. Below this, it shows 'IAM resources' with counts: 'Users: 4', 'Groups: 1', 'Roles: 37', 'Identity providers: 0', and 'Customer managed policies: 3'. A 'Best practices' section lists several recommendations, including 'Grant least privilege access', 'Enable Identity federation', 'Enable MFA', 'Rotate credentials', and 'Enable IAM Access Analyzer'. A 'What's new' section at the bottom mentions 'Learn about the latest releases for AWS Identity & Access Management (IAM)'. The right sidebar contains 'Additional information' with links to 'IAM documentation' and 'Videos, IAM release history and additional resources', and 'Tools' with links to 'Web identity federation playground' and 'Policy simulator'. A 'Quick links' section at the bottom right contains a link to 'My access key'.

aws Services ▼

Identity and Access Management (IAM)

Dashboard

▼ Access management

- Groups
- Users
- Roles**
- Policies
- Identity providers
- Account settings

▼ Access reports

- Access analyzer
- Archive rules
- Analyzers
- Settings
- Credential report
- Organization activity
- Service control policies (SCPs)

## IAM dashboard

Sign-in URL for IAM users in this account

<https://754261393455.signin.aws.amazon.com/console> | [Customize](#)

### IAM resources

Users: 4	Roles: 37
Groups: 1	Identity providers: 0
Customer managed policies: 3	

### Best practices

- Grant [least privilege access](#): Establishing a principle of least privilege ensures that identities are only permitted to perform the most minimal set of functions necessary to fulfill a specific task, while balancing usability and efficiency.
- Enable Identity federation: Centrally manage users and access across multiple applications and services. For federation to multiple accounts in your AWS Organization, you can configure your identity source in [AWS Single Sign-on](#).
- Enable MFA: For extra security, we recommend that you require multi-factor authentication (MFA) for [all users](#).
- [Rotate credentials](#) regularly: Change your own passwords and access keys regularly, and make sure that all users in your account do as well.
- Enable [IAM Access Analyzer](#): Enable IAM Access Analyzer to analyze public, cross-account, and cross-organization access.

Learn more about all [security best practices](#)

### What's new

Learn about the latest releases for AWS Identity & Access Management (IAM)

### Additional information

- [IAM documentation](#)
- [Videos, IAM release history and additional resources](#)

### Tools

- [Web identity federation playground](#)
- [Policy simulator](#)

### Quick links

- [My access key](#)

# Create Role

Browser address bar: <https://console.aws.amazon.com/iam/home?region=us-west-1#/roles>

Navigation bar: Create SaaS Contracts ... AWS Marketplace: Clo... | aws Services | Pascal.j @ 7542-6139-3455 | Global | Support

**Identity and Access Management (IAM)**

- Dashboard
- Access management
  - Groups
  - Users
  - Roles**
  - Policies
  - Identity providers
  - Account settings
- Access reports
  - Access analyzer
    - Archive rules
    - Analyzers
    - Settings
  - Credential report
  - Organization activity
  - Service control policies (SCPs)

**Roles**

**What are IAM roles?**

IAM roles are a secure way to grant permissions to entities that you trust. Examples of entities include the following:

- IAM user in another account
- Application code running on an EC2 instance that needs to perform actions on AWS resources
- An AWS service that needs to act on resources in your account to provide its features
- Users from a corporate directory who use identity federation with SAML

IAM roles issue keys that are valid for short durations, making them a more secure way to grant access.

**Additional resources:**

- [IAM Roles FAQ](#)
- [IAM Roles Documentation](#)
- [Tutorial: Setting Up Cross Account Access](#)
- [Common Scenarios for Roles](#)

**Create role** **Delete role**

Search Showing 37 results

Role name	Trusted entities	Last activity
-----------	------------------	---------------

# Select EC2 Use case, click Next: Permissions

aws

Services ▾

Pascal.j @ 7542-6139-3455 ▾

Global ▾

Support ▾

**AWS service**  
EC2, Lambda and others

**Another AWS account**  
Belonging to you or 3rd party

**Web identity**  
Cognito or any OpenID provider

**SAML 2.0 federation**  
Your corporate directory

Allows AWS services to perform actions on your behalf. [Learn more](#)

### Choose a use case

**Common use cases**

**EC2**  
Allows EC2 instances to call AWS services on your behalf.

**Lambda**  
Allows Lambda functions to call AWS services on your behalf.

**Or select a service to view its use cases**

<a href="#">API Gateway</a>	<a href="#">CloudWatch Events</a>	<a href="#">EKS</a>	<a href="#">KMS</a>	<a href="#">Rekognition</a>
<a href="#">AWS Backup</a>	<a href="#">CodeBuild</a>	<a href="#">EMR</a>	<a href="#">Kinesis</a>	<a href="#">RoboMaker</a>
<a href="#">AWS Chatbot</a>	<a href="#">CodeDeploy</a>	<a href="#">ElastiCache</a>	<a href="#">Lake Formation</a>	<a href="#">S3</a>
<a href="#">AWS Marketplace</a>	<a href="#">CodeGuru</a>	<a href="#">Elastic Beanstalk</a>	<a href="#">Lambda</a>	<a href="#">SMS</a>
<a href="#">AWS Support</a>	<a href="#">CodeStar Notifications</a>	<a href="#">Elastic Container Service</a>	<a href="#">Lex</a>	<a href="#">SNS</a>
<a href="#">Amplify</a>	<a href="#">Comprehend</a>	<a href="#">Elastic Transcoder</a>	<a href="#">License Manager</a>	<a href="#">SWF</a>
<a href="#">AppStream 2.0</a>	<a href="#">Config</a>	<a href="#">ElasticLoadBalancing</a>	<a href="#">Machine Learning</a>	<a href="#">Step Functions</a>

\* Required

Cancel







Next: Permissions

# Select AmazonS3FullAccess Policy

Filter policies ▾

Q s3

Showing 6 results

	Policy name ▾	Used as
<input type="checkbox"/>	 AmazonDMSRedshiftS3Role	None
<input checked="" type="checkbox"/>	 AmazonS3FullAccess	None
<input type="checkbox"/>	 AmazonS3OutpostsFullAccess	None
<input type="checkbox"/>	 AmazonS3OutpostsReadOnlyAccess	None
<input type="checkbox"/>	 AmazonS3ReadOnlyAccess	None
<input type="checkbox"/>	 QuickSightAccessForS3StorageManagementAnalyticsReadOnly	None



# Name your Role S3\_Access and Create role

## Create role



### Review

Provide the required information below and review this role before you create it.

**Role name\***

S3\_Access

Use alphanumeric and '+=, @-\_' characters. Maximum 64 characters.

**Role description**

Allows EC2 instances to call AWS services on your behalf.

Maximum 1000 characters. Use alphanumeric and '+=, @-\_' characters.

**Trusted entities**

AWS service: ec2.amazonaws.com

**Policies**



AmazonS3FullAccess [↗](#)

**Permissions boundary**

Permissions boundary is not set

*No tags were added.*

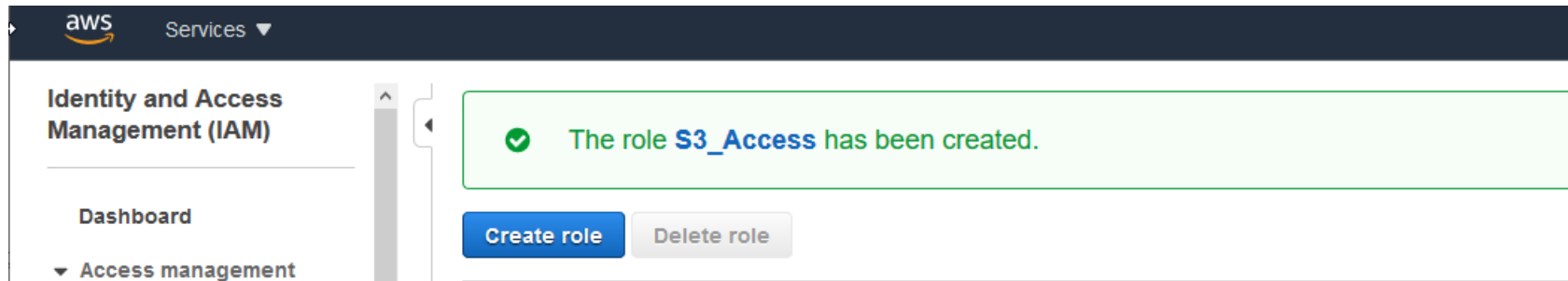
\* Required

Cancel

Previous

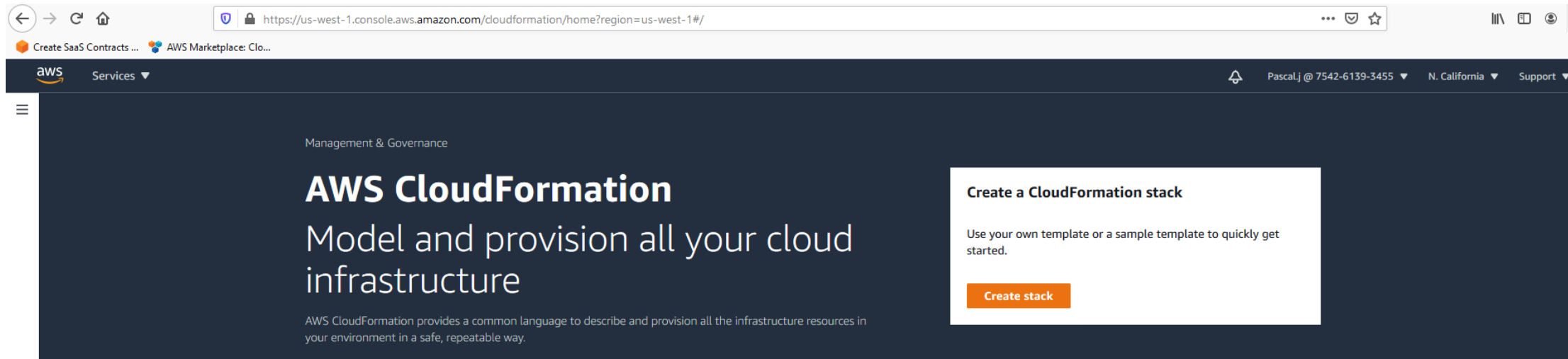
Create role

# This role will be an input of your sandbox



# generate the S3 role using a CF template

Navigate to Services: Cloud Formation to Create Stack



# Upload template file *s3\_bucket\_access.template* provided in the workshop

aws Services ▾

CloudFormation > Stacks > Create stack

Step 1  
**Specify template**

Step 2  
Specify stack details

Step 3  
Configure stack options

Step 4  
Review

## Create stack

### Prerequisite - Prepare template

Prepare template  
Every stack is based on a template. A template is a JSON or YAML file that contains configuration information about the AWS resources you want to include in the stack.

☒ Template is ready ☐ Use a sample template ☐ Create template in Designer

### Specify template

A template is a JSON or YAML file that describes your stack's resources and properties.

Template source  
Selecting a template generates an Amazon S3 URL where it will be stored.

☐ Amazon S3 URL ☒ Upload a template file

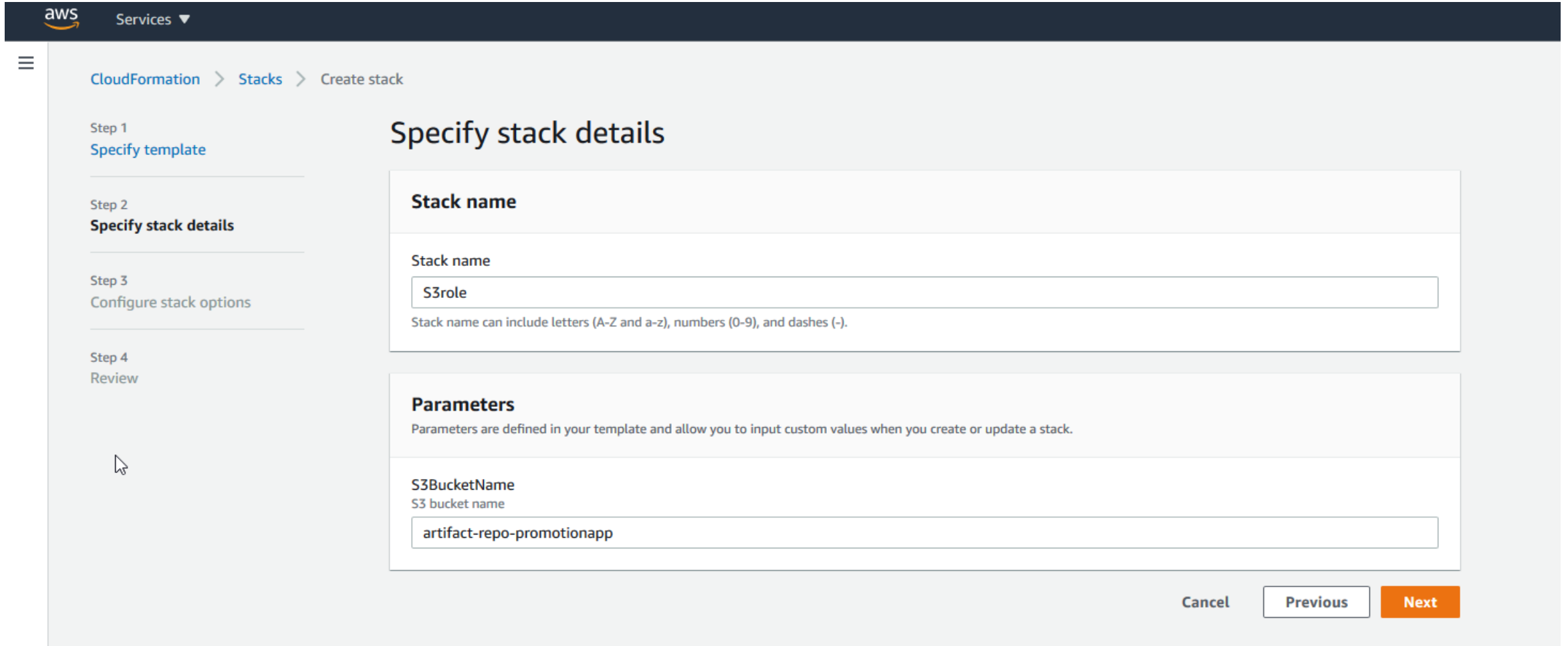
Upload a template file

*s3\_bucket\_access.template*

JSON or YAML formatted file

S3 URL: [https://s3-us-west-1.amazonaws.com/cf-templates-jq3r56jlrdbw-us-west-1/2020288PHT-s3\\_bucket\\_access.template](https://s3-us-west-1.amazonaws.com/cf-templates-jq3r56jlrdbw-us-west-1/2020288PHT-s3_bucket_access.template)

# Enter a name for the stack and specify the S3 bucket name as input parameter



The screenshot shows the AWS CloudFormation console interface. At the top, the AWS logo and 'Services' dropdown are visible. The breadcrumb navigation indicates the path: CloudFormation > Stacks > Create stack. On the left, a sidebar lists four steps: Step 1 (Specify template), Step 2 (Specify stack details), Step 3 (Configure stack options), and Step 4 (Review). Step 2 is currently selected and highlighted. The main content area is titled 'Specify stack details' and contains two sections: 'Stack name' and 'Parameters'. In the 'Stack name' section, the input field contains 'S3role'. In the 'Parameters' section, the input field for 'S3BucketName' contains 'artifact-repo-promotionapp'. At the bottom right, there are three buttons: 'Cancel', 'Previous', and 'Next'.

aws Services ▼

CloudFormation > Stacks > Create stack

Step 1  
Specify template

Step 2  
**Specify stack details**

Step 3  
Configure stack options

Step 4  
Review

## Specify stack details

**Stack name**

Stack name

S3role

Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).

**Parameters**

Parameters are defined in your template and allow you to input custom values when you create or update a stack.

S3BucketName  
S3 bucket name

artifact-repo-promotionapp

Cancel Previous Next

# Follow the Next screens until you get to the stack creation button (check box at the bottom)

All other values can be kept default

► [Quick-create link](#)

Capabilities


**i** The following resource(s) require capabilities: [AWS::IAM::Role]


This template contains Identity and Access Management (IAM) resources that might provide entities access to make changes to your AWS account. Check that you want to create each of these resources and that they have the minimum required permissions. [Learn more](#)

☒ I acknowledge that AWS CloudFormation might create IAM resources.

Cancel Previous Create change set Create stack

# S3 role is now created. This role will be an input of your Jenkins sandbox

 Services ▾

 Pascal.Lj @ 7542-6139-3455 ▾ N. California ▾ Support ▾

CloudFormation > Stacks > S3role

Stacks (1)


Filter by stack name

Active ▾ ☒ View nested

< 1 >

S3role

2020-10-14 10:40:47 UTC-0700

 CREATE\_COMPLETE

S3role

Delete Update Stack actions ▾ Create stack ▾

Stack info Events Resources **Outputs** Parameters Template Change sets

Outputs (1)

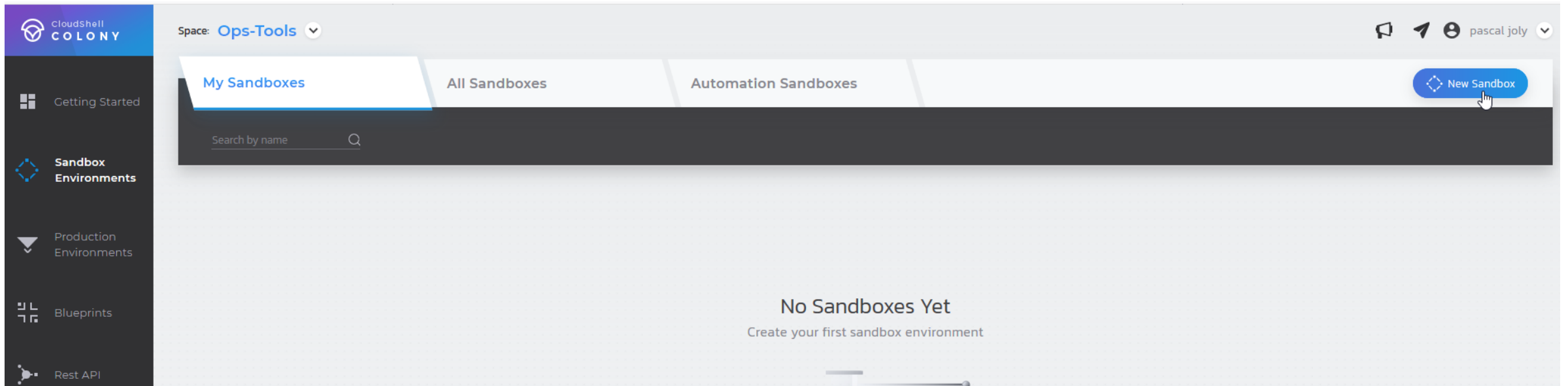
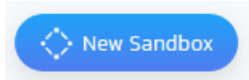
Search outputs

Key	Value	Description	Export name
IAMRoleInstanceProfile	S3role-ColonyJenkinsInstance2Profile-RVS801684YR6	IAM Role Instance Profile name	-

# Deploy Jenkins Sandbox in Colony

Navigate to the Sandbox Environment of the OpsTools space in Colony.

Click on






# Select Jenkins blueprint and click Next

Launch a New Sandbox

Select Blueprint    Set Parameters

Space Blueprints (1)    Samples (4)

Select a Blueprint: Search by name

**Jenkins**  
1 App | AWS

Blueprint that deploys Jenkins in AWS

[More info](#)

[Cancel](#) [Next: Set Parameters](#)


# Enter required parameters as below

Launch a New Sandbox

Select Blueprint

Set Parameters

Blueprint: Jenkins [Change Blueprint](#)

Cloud Provider:  AWS

GENERAL

\* Sandbox Name

Jenkins\_CI

Duration ⓘ

0

0

2

0

Weeks

Days

Hours

Minutes

PARAMETERS

\* AWS\_INSTANCE\_TYPE ⓘ

m5.large

\* CS\_COLONY\_TOKEN ⓘ

.....

\* SPACE\_NAME ⓘ

Trial

\* JENKINS\_ADMIN\_PASS ⓘ

.....

\* IAM\_ROLE ⓘ

lonyJenkinsInstance2Profile-RVS801684YR6

\* BUCKET\_NAME ⓘ

artifact-repo-promotionapp

Cancel

Launch Sandbox

# Launch Sandbox

Launch a New Sandbox

1


Select Blueprint

2

Set Parameters

You just launched your first sandbox. Congratulations!


Blueprint: Jenkins

Cloud Provider:  AWS

Parameters: Sandbox Name Jenkins CI

**Congratulations your sandbox is being launched!**

Open the sandbox to track the launch progress



Open Sandbox >

Help

# Once the sandbox deployment is complete, click on the web “quick link” to access Jenkins

The screenshot displays the Cloudshell COLONY interface for a deployment named 'Jenkins\_CI' in the 'Ops-Tools' space. The deployment is 'Active' and was launched on Oct 13, 2020 at 4:17 PM, with a deployment duration of 6 minutes. The 'SUMMARY' tab is selected, showing the 'Jenkins' blueprint (AWS US West (N. California)) and its parameters. A 'QUICK LINKS (1)' panel on the right provides a direct URL to access Jenkins. The 'TIMELINE' section shows an up time of 12m 16s and an end time of Oct 13, 2020 8:16 PM. The 'OWNER' is listed as pascal.joly.

**Cloudshell COLONY**

Space: **Ops-Tools**

**Jenkins\_CI** Active Cloud Provider: AWS US West (N. California) 3h 46m Remaining Extend End

**Summary** Applications Infrastructure Troubleshooting

**ACTIVE** (1 Apps)

**Jenkins\_CI**  
Launched at: Oct 13, 2020 4:17 PM  
Deployment duration: 6 minutes

**BLUEPRINT**

**Jenkins**  
AWS US West (N. California)  
Blueprint that deploys Jenkins in AWS  
[More info](#)

**SANDBOX PARAMETERS**

AWS_INSTANCE_TYPE	m5.large
CS_COLONY_TOKEN	*****
SPACE_NAME	Trial
JENKINS_ADMIN_PASS	*****
IAM_ROLE	S3_Access

**QUICK LINKS (1)**

- jenkins  
<http://sandr-mainA-1XZ45Y03HEF39-1784920524.us-west-1.elb.amazonaws.com:8080>

**Help**

# Login to the console

← → ↻ 🏠 | 🔒 sandb-maina-ijdp3m33i4c-884482574.us-west-1.elb.amazonaws.com:8080 | ⋮ 🛡️ ☆ | ⚡ 📄 🗑️ 🌙 ☰

Create SaaS Contracts ... AWS Marketplace: Clo...

## Jenkins

🔍 search ? **log in**

Jenkins ▶

👤 People

📝 Build History

🌊 Open Blue Ocean

🔑 Credentials

**Build Queue** —

No builds in the queue.

**Build Executor Status** —

1 Idle

2 Idle

Jenkins Test Colony Installation

All

S	W	Name ↓	Last Success	Last Failure	Last Duration
🌐	☀️	<a href="#">demo</a>	N/A	N/A	N/A

Icon: [S](#) [M](#) [L](#)

[Legend](#) 📡 [Atom feed for all](#) 📡 [Atom feed for failures](#) 📡 [Atom feed for just latest builds](#)


# Use the default “admin” password if you did not change it in the sandbox parameters



Welcome to Jenkins!


Sign in


☒ Keep me signed in


 **Jenkins**

search

?

 4

 admin

 log out

Jenkins

New Item

People

Build History

Manage Jenkins

My Views

Open Blue Ocean





New View

Jenkins Test Colony Installation

add description


All


+


S	W	Name ↓	Last Success	Last Failure	Last Duration	Fav
		<a href="#">demo</a>	N/A	N/A	N/A	 

Icon: [S](#) [M](#) [L](#)

Legend

 [Atom feed for all](#)

 [Atom feed for failures](#)

 [Atom feed for just latest builds](#)

Build Queue

No builds in the queue.

Build Executor Status

1 Idle

2 Idle



# Deploy a sandbox from Jenkins.

- Specify S3 artifact repo (as a parameter to Jenkins config). This is the repository that was configured in Module 1
- Run the Jenkins Job:
  - Deploy latest version of promotion app
  - Retrieve environment details (URL end point) and outputs it to the console to feed it to a test (test is a placeholder step).
  - Terminate the sandbox
- More info:  
<https://quali.atlassian.net/wiki/spaces/BD/pages/809500683/Jenkins+solution+for+Colony>
- <https://quali.atlassian.net/wiki/spaces/BD/pages/875036714/How+to+handle+artifacts>

# Click on the pipeline demo to update the name of the blueprint that will be deployed

Jenkins Test Colony Installation

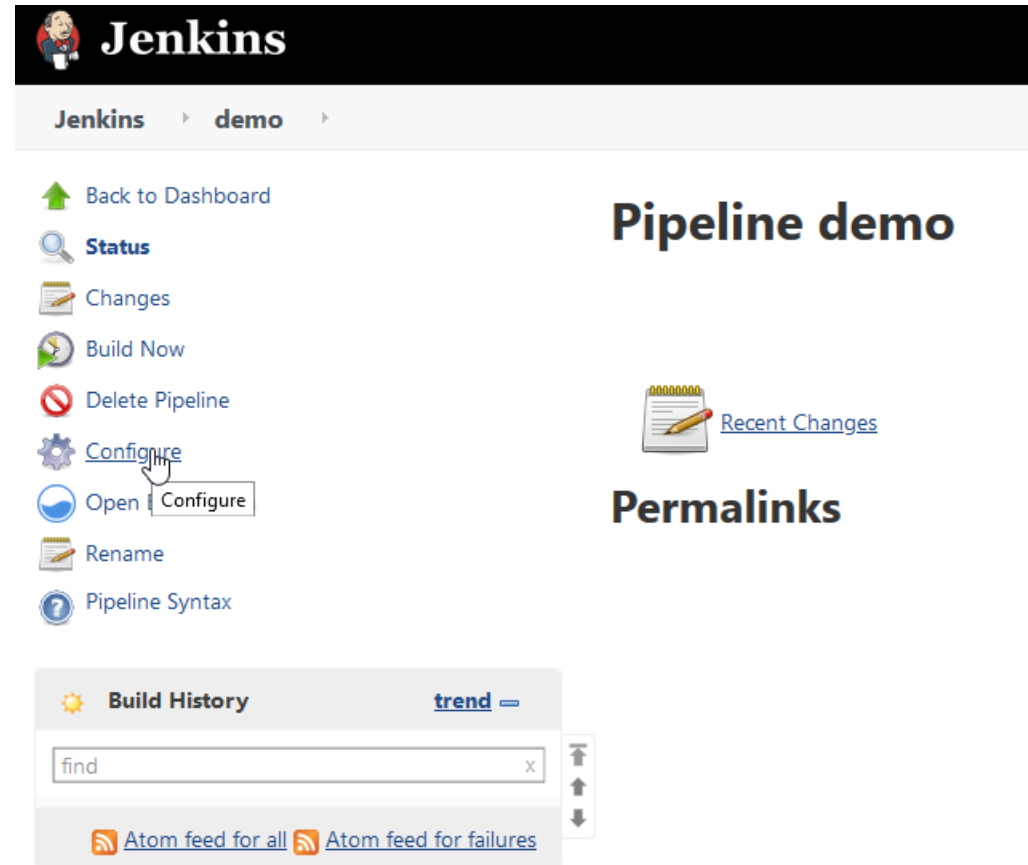
All +

S	W	Name ↓	Last Success	Last Failure
		<a href="#">demo</a> ▼	N/A	N/A

Icon: [S](#) [M](#) [L](#)












# Click Configure




**Jenkins**



Jenkins > demo >


-  [Back to Dashboard](#)
-  [Status](#)
-  [Changes](#)
-  [Build Now](#)
-  [Delete Pipeline](#)
-  [Configure](#)
-  [Open | Configure](#)
-  [Rename](#)
-  [Pipeline Syntax](#)



**Pipeline demo**

 [Recent Changes](#)

**Permalinks**

 **Build History** [trend](#) 



 [Atom feed for all](#)  [Atom feed for failures](#)

# Update the blueprint name to promotion-manager-aws (or any variation of it from Module 1 available in your Trial space)

## Pipeline

Definition Pipeline script

```
Script 74 //Start Sandbox
75 //TODO replace latest with current artifact from Build stage
76 newSandboxData = colony.blueprint(
77     spaceName, // space name
78     'Promotions Manager AWS', // blueprint name
79     'pm-test-sandbox-' + env.BUILD_NUMBER, // sandbox name
80     'PT2H', // sandbox duration according to ISO 8601 duration
81     ['promotions-manager-ui': 'promotions-manager/latest/promotions-manager-ui.master.tar.gz', // artifacts
82      'promotions-manager-api': 'promotions-manager/latest/promotions-manager-api.master.tar.gz',
83      'mongodb': 'promotions-manager/latest/test-db.tar.gz'],
```

## Pipeline

Definition Pipeline script

```
Script 74 //Start Sandbox
75 //TODO replace latest with current artifact from Build stage
76 newSandboxData = colony.blueprint(
77     spaceName, // space name
78     'promotions-manager-all-aws', // blueprint name
79     'pm-test-sandbox-' + env.BUILD_NUMBER, // sandbox name
80     'PT2H', // sandbox duration according to ISO 8601 duration
81     ['promotions-manager-ui': 'promotions-manager/latest/promotions-manager-ui.master.tar.gz', // artifacts
82      'promotions-manager-api': 'promotions-manager/latest/promotions-manager-api.master.tar.gz',
83      'mongodb': 'promotions-manager/latest/test-db.tar.gz'],
84     ['PORT': '3000', 'API_PORT': '3001', 'RELEASE_NUMBER': 'none', 'API_BUILD_NUMBER': env.BUILD_NUMBER, 'AWS_INSTANCE_TYPE': 't3.large'
85     30) // timeout
86     .startSandbox()
87
88     def endpoint = getEndpoint(newSandboxData)
89
90     --
```

☒ Use Groovy Sandbox

[Pipeline Syntax](#)

# Pipeline steps

The default pipeline loaded with this Jenkins instance will automatically run the following steps:

1. Retrieve latest code of the promotion app from Git
2. Build artifact package (latest build)
3. Upload Build to S3 repository
4. Deploy the promotion application to AWS in a Colony Sandbox
5. Run tests
6. Terminate the promotion application

# Run the pipeline: BuildNow

Jenkins

demo

Back to Dashboard

Status

Changes

Build Now

Build Now

Configure

Open Blue Ocean

Rename

Pipeline Syntax

## Pipeline demo

[Recent Changes](#)

## Permalinks

Build History

[trend](#)


#1

Oct 13, 2020 11:37 PM





[Atom feed for all](#)

[Atom feed for failures](#)


# Check results/console Output


 **Jenkins**


search


  4  admin  log out


Jenkins > demo > #1


 Back to Project


 Status


 Changes


 **Console Output**


 View as plain text


 Edit Build Information


 Delete build '#1'


 Git Build Data


 No Tags


 Git Build Data


 Git Build Data


 Open Blue Ocean

 Restart from Stage

 Replay

 Pipeline Steps

 Workspaces

 **Console Output**

Started by user [admin](#)

Running in Durability level: MAX\_SURVIVABILITY

[Pipeline] Start of Pipeline

[Pipeline] node

Running on [Jenkins](#) in /var/jenkins\_home/jobs/demo/workspace

[Pipeline] {

[Pipeline] stage

[Pipeline] { (Checkout)

[Pipeline] dir

Running in /var/jenkins\_home/jobs/demo/workspace/UI

[Pipeline] {

[Pipeline] git

The recommended git tool is: NONE

No credentials specified

Cloning the remote Git repository

Cloning repository [https://QualiLabs@dev.azure.com/QualiLabs/Promotions%20Manager/\\_git/promotions-manager-ui](https://QualiLabs@dev.azure.com/QualiLabs/Promotions%20Manager/_git/promotions-manager-ui)

> git init /var/jenkins\_home/jobs/demo/workspace/UI # timeout=10

Fetching upstream changes from [https://QualiLabs@dev.azure.com/QualiLabs/Promotions%20Manager/\\_git/promotions-manager-ui](https://QualiLabs@dev.azure.com/QualiLabs/Promotions%20Manager/_git/promotions-manager-ui)

> git --version # timeout=10

> git --version # 'git version 2.11.0'

> git fetch --tags --progress -- [https://QualiLabs@dev.azure.com/QualiLabs/Promotions%20Manager/\\_git/promotions-manager-ui](https://QualiLabs@dev.azure.com/QualiLabs/Promotions%20Manager/_git/promotions-manager-ui) +refs/heads/\*:refs/remotes/origin/\* # timeout=10

> git config remote.origin.url [https://QualiLabs@dev.azure.com/QualiLabs/Promotions%20Manager/\\_git/promotions-manager-ui](https://QualiLabs@dev.azure.com/QualiLabs/Promotions%20Manager/_git/promotions-manager-ui) # timeout=10

> git config --add remote.origin.fetch +refs/heads/\*:refs/remotes/origin/\* # timeout=10

Avoid second fetch

> git rev-parse refs/remotes/origin/master^{commit} # timeout=10

Checking out Revision 3945c86cc0a381c43075210c15804cb60d37b86f (refs/remotes/origin/master)

> git config core.sparsecheckout # timeout=10

> git checkout -f 3945c86cc0a381c43075210c15804cb60d37b86f # timeout=10

> git branch -a -v --no-abbrev # timeout=10

> git checkout -b master 3945c86cc0a381c43075210c15804cb60d37b86f # timeout=10

Commit message: "Updated Jenkinsfile"

First time build. Skipping changelog.

# Check the sandbox deployment in the output

```
Starting sandbox...
[Pipeline] echo
health check - waiting for sandbox h50auo0rpv02c1 to become ready for testing...
[Pipeline] waitForSandbox
[Pipeline] echo
health check done! returned:{
  "blueprint_description": "All in one deployment of our promotions manager\n",
  "applications": [
    {
      "private_address": "10.0.0.229",
      "internal_ports": [
        "27017"
      ],
      "external_ports": [],
      "shortcuts": [],
      "image_name": "ami-08660f1c6fb6b01e7",
      "internal_dns": "mongodb.h50auo0rpv02c1.sandbox.com",
      "instances": [
        {
          "status": "Done",
          "infra_id": "i-0f9ca47e5ce316861",
          "instance_type": "t3.large",
          "access_links": [],
          "private_ip": "10.0.0.229",
          "compute_availability": "on_demand"
        }
      ]
    }
  ]
}
```

# Check that the sandbox was automatically created in Colony

- Browse to Automation Sandboxes tab in the Trial space

The screenshot shows the CloudShell COLONY interface. On the left is a dark sidebar with navigation options: 'Getting Started', 'Sandbox Environments' (highlighted with a blue diamond icon), and 'Production Environments'. The main area has a top header with 'Space: Trial' and a user profile 'pascal joly'. Below the header are three tabs: 'My Sandboxes', 'All Sandboxes', and 'Automation Sandboxes' (which is selected and highlighted with a blue border). A 'New Sandbox' button is in the top right. A search bar labeled 'Search by name' is below the tabs. A table lists the automation sandboxes with columns: STATUS, SANDBOX NAME, START, END, CLOUD, BLUEPRINT, and OWNER. One entry is visible: 'pm-test-sandbox-1' with status 'Ended', created '3 hours ago', on 'AWS EU (Ireland)', with blueprint 'promotions-manager-all-aws', and owner 'pascal joly'.

STATUS	SANDBOX NAME	START	END	CLOUD	BLUEPRINT	OWNER
Ended	pm-test-sandbox-1	3 hours ago	3 hours ago	AWS EU (Ireland)	promotions-manager-all-aws	pascal joly







# Learn More

CloudShell Colony offers many other plugins with CI tools, check the [Colony integration](#) website for more details

Quali SOLUTIONS PRODUCTS PRICING RESOURCES COMPANY PARTNERS CONTACT A SPECIALIST ⓘ 🔍

CloudShell COLONY

FEATURED INTEGRATIONS

 Jenkins DOWNLOAD PLUG IN LEARN MORE	 Azure DevOps DOWNLOAD PLUG IN LEARN MORE	 TeamCity DOWNLOAD PLUG IN LEARN MORE
 Bamboo DOWNLOAD PLUG IN LEARN MORE	 Circleci DOWNLOAD PLUG IN LEARN MORE	 Artifactory LEARN MORE