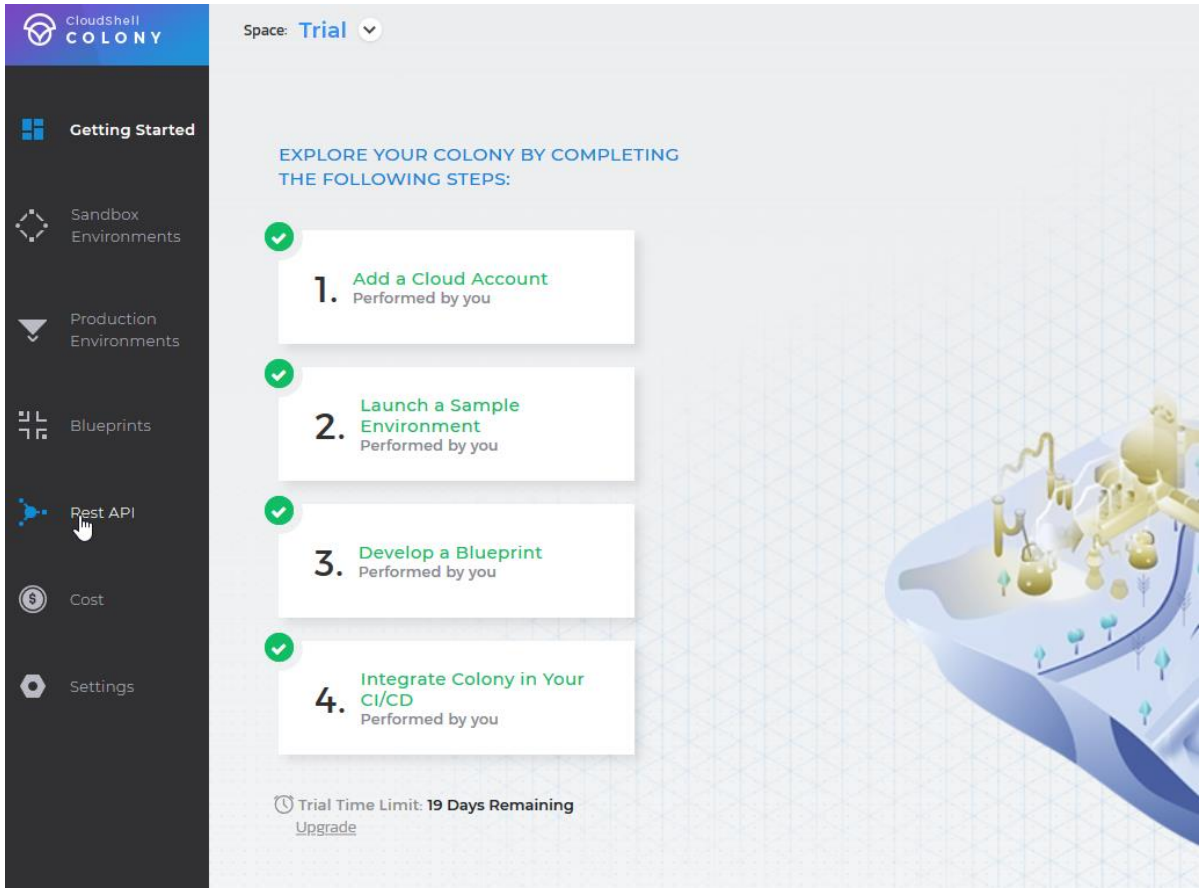
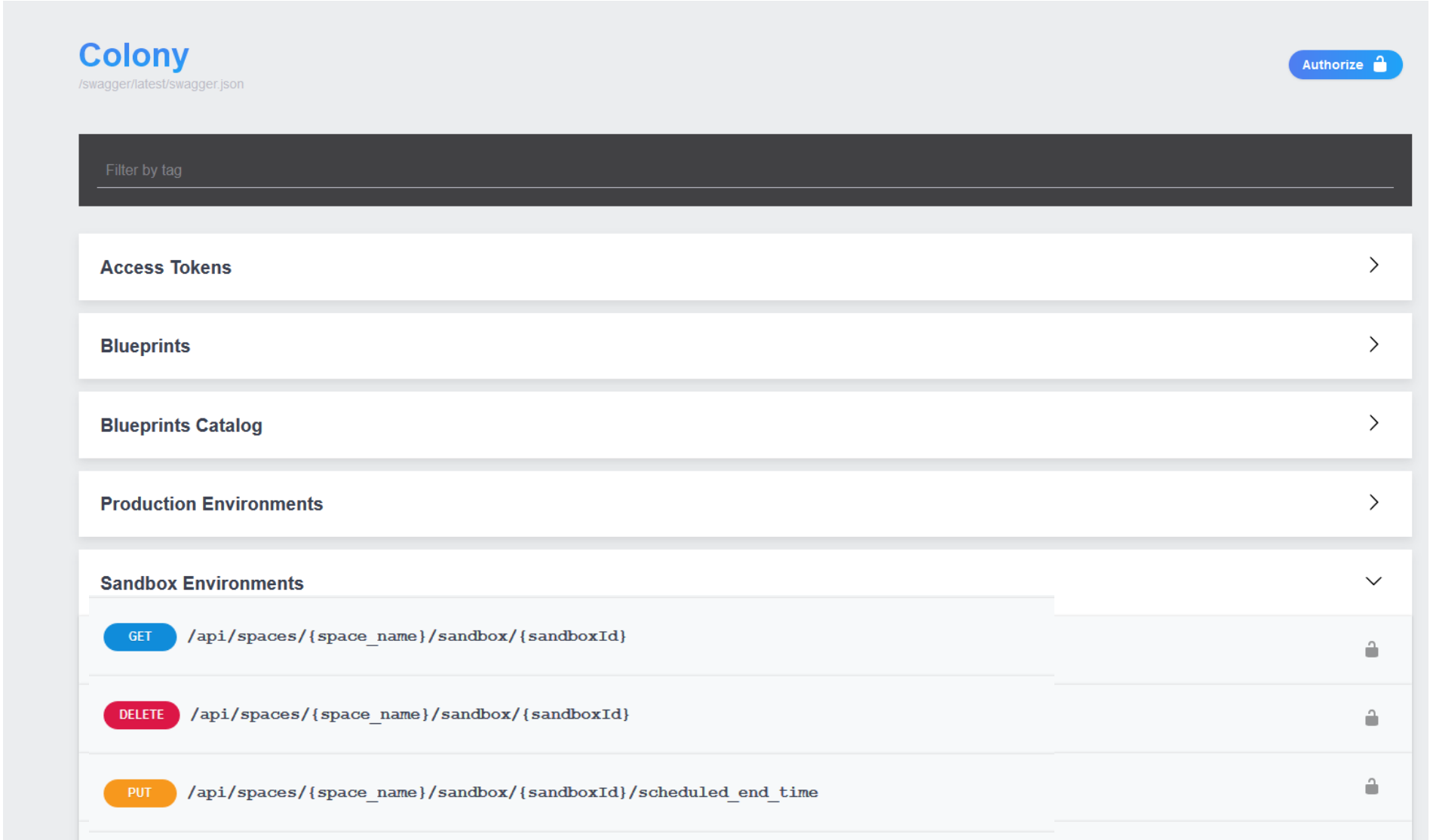
Module 3

- 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

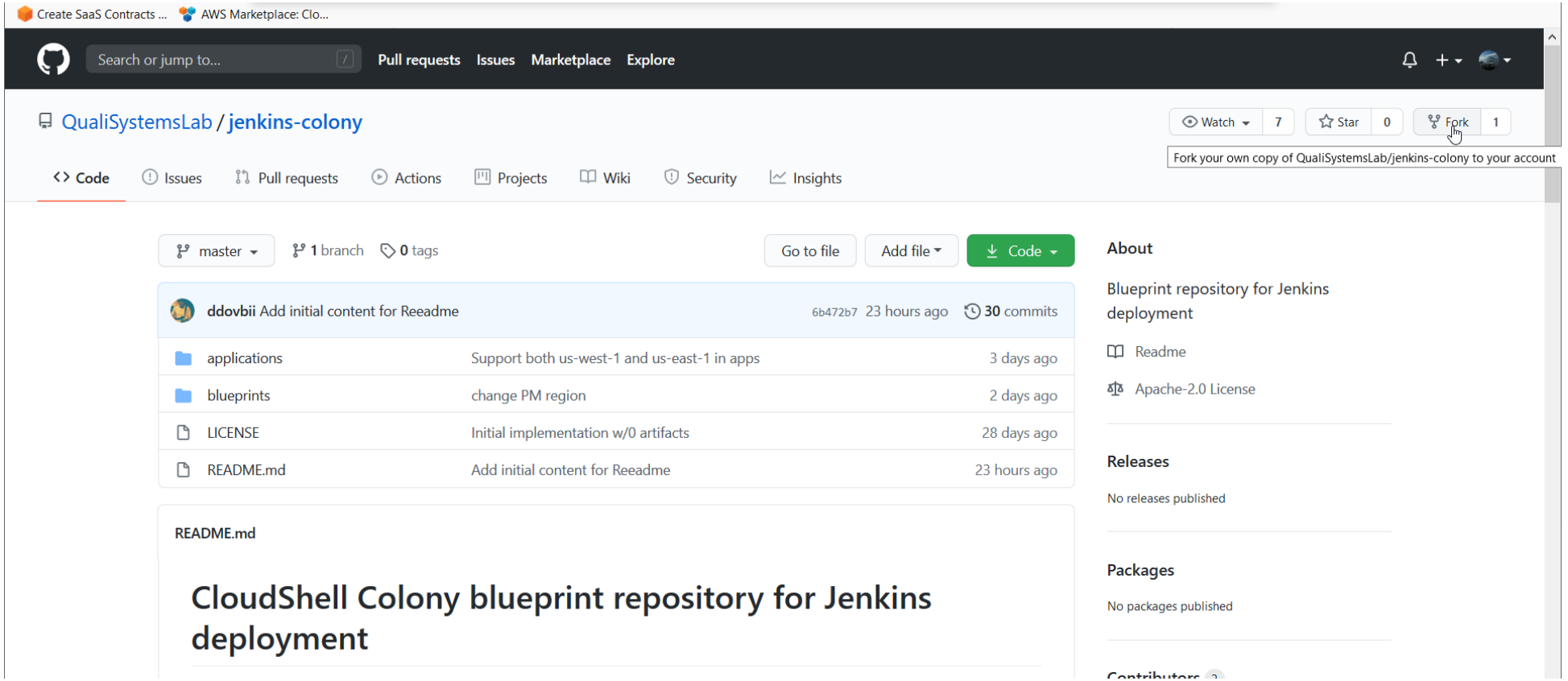


Deploy and Configure Jenkins

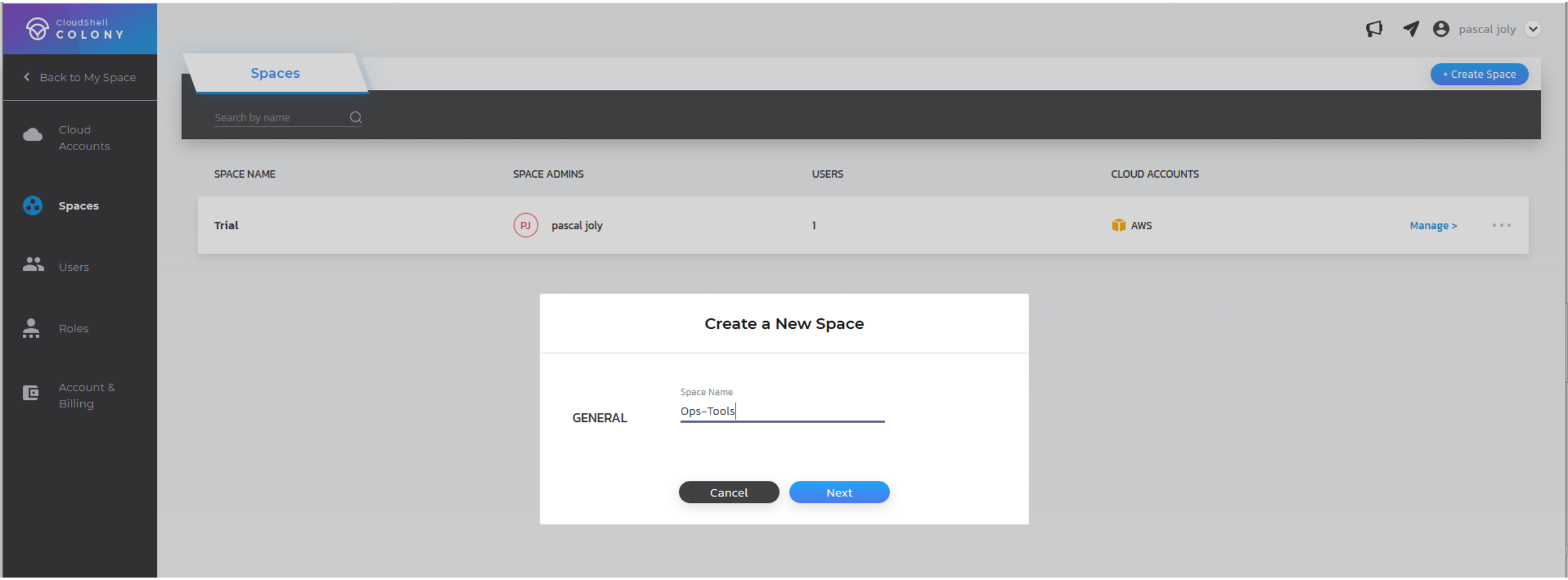
- The first step will be for you to automatically deploy and configure Jenkins using CloudShell Colony.
- This step would typically be performed by a platform Ops/Tools team ahead of time.

Fork the repository with the Jenkins Blueprint

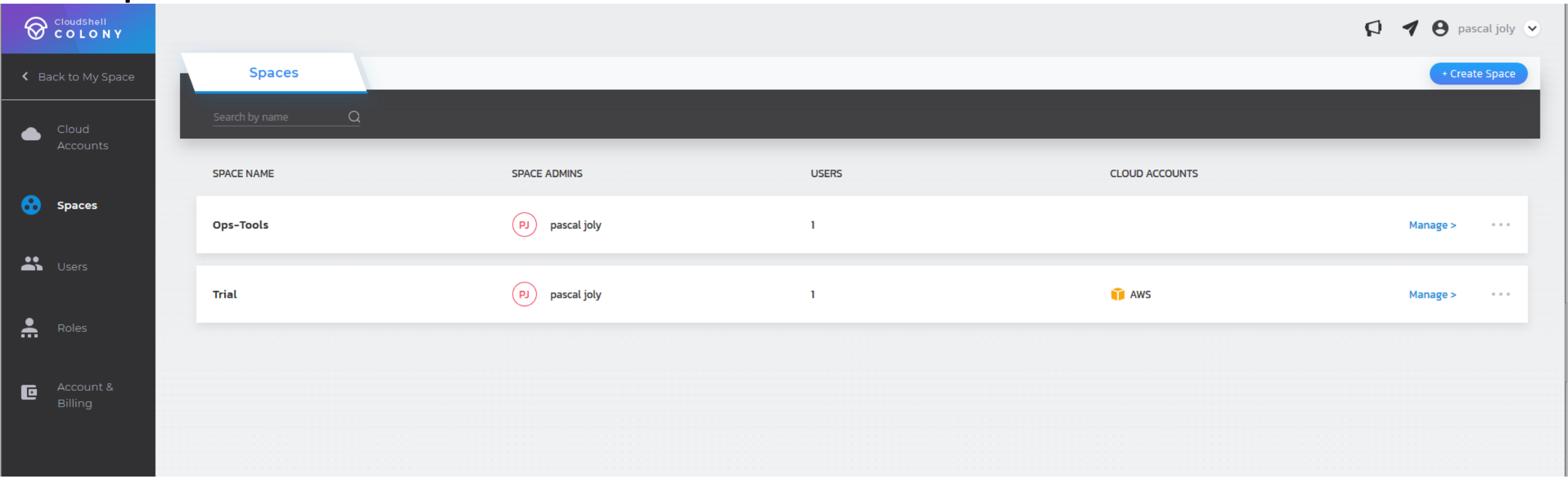
- https://github.com/QualiSystemsLab/jenkins-colony



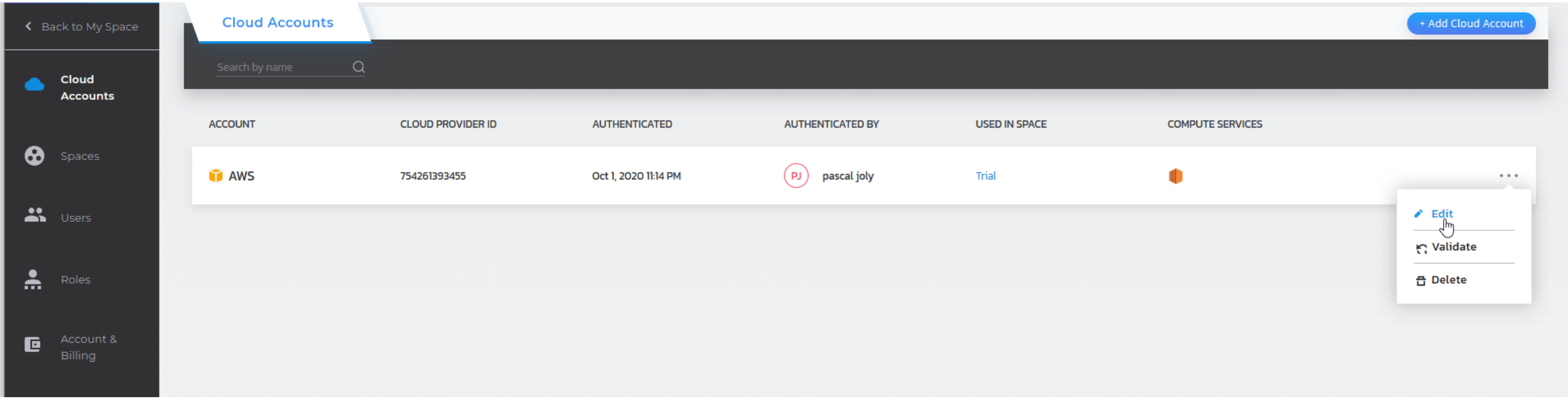
Create a new “Ops-Tools” space in Colony

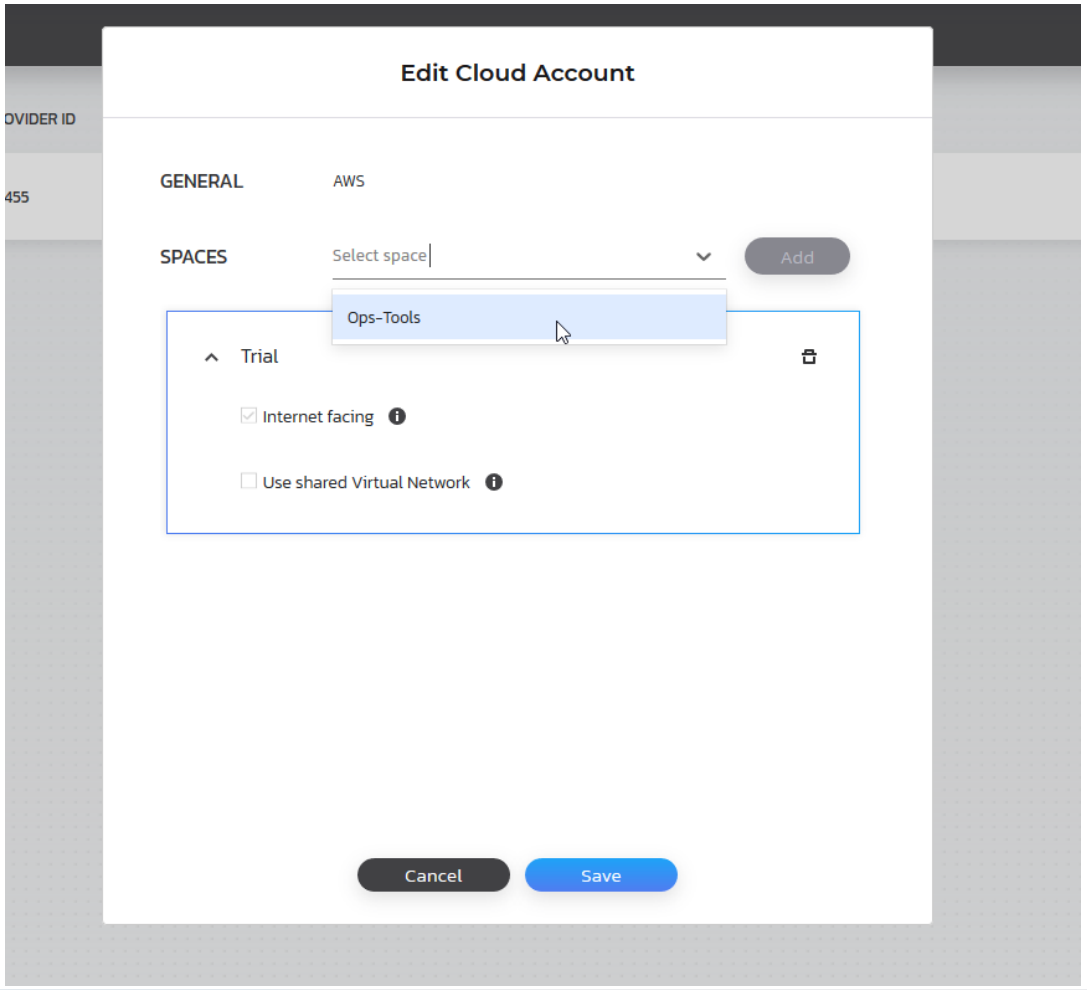


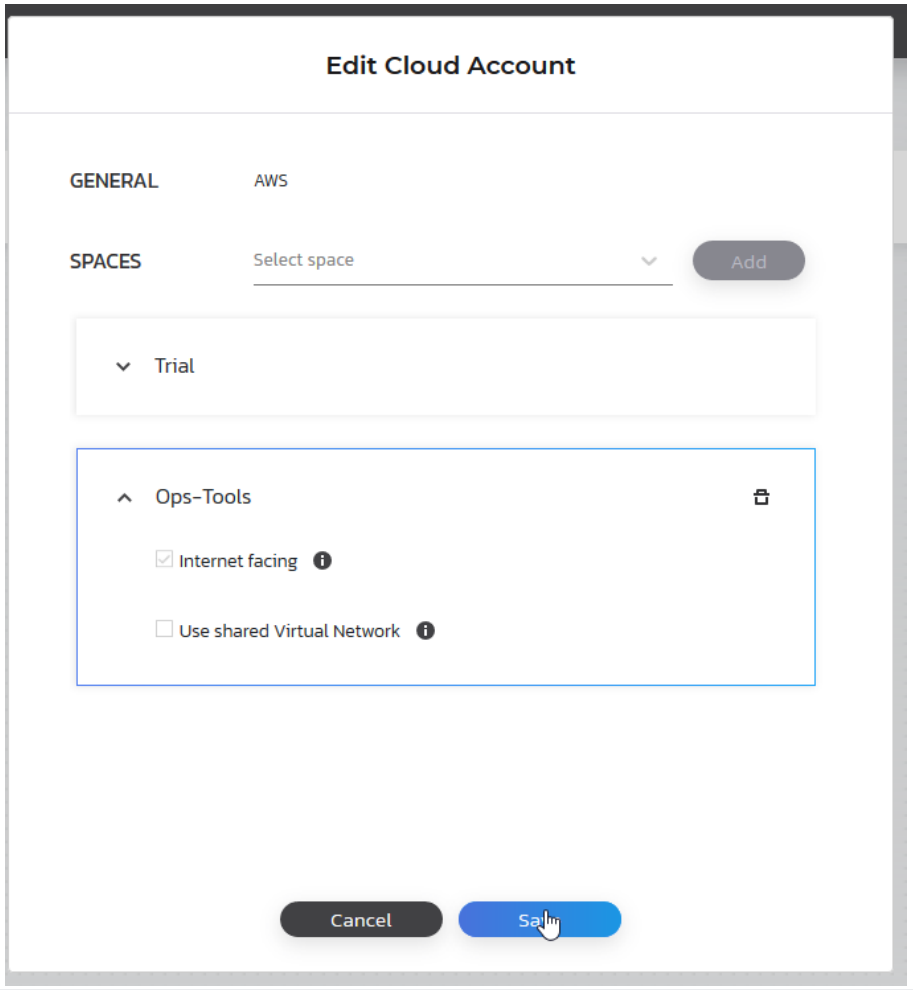
You will use this space to manage Ops platforms such as Jenkins

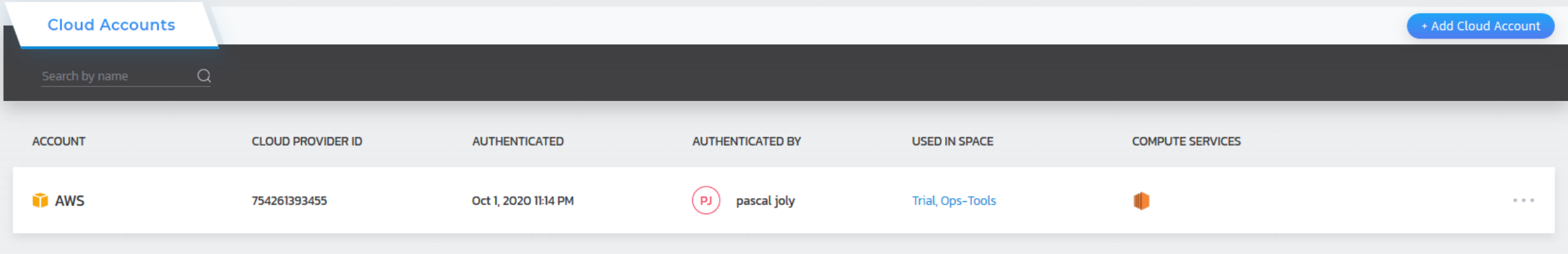


Add your AWS cloud account to the Ops-Tools space

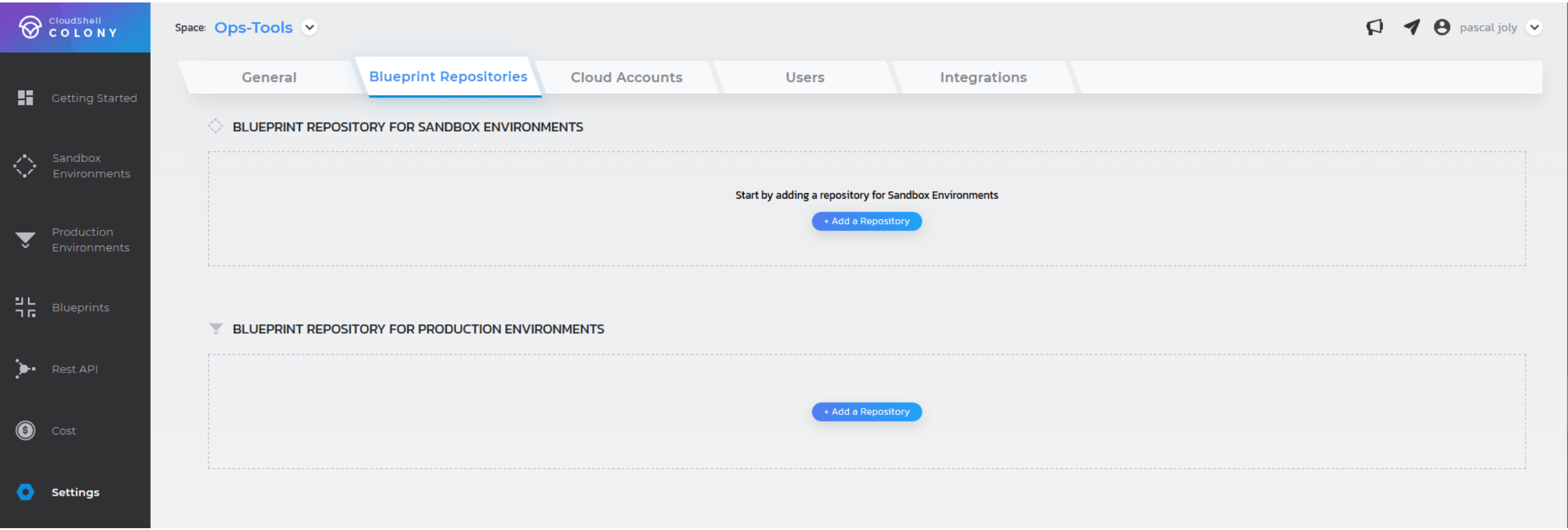



Edit Cloud Account
GENERAL
AWS
SPACES
Select space
Add
Ops-Tools
Trial
Internet facing
Use shared Virtual Network
Cancel
Save

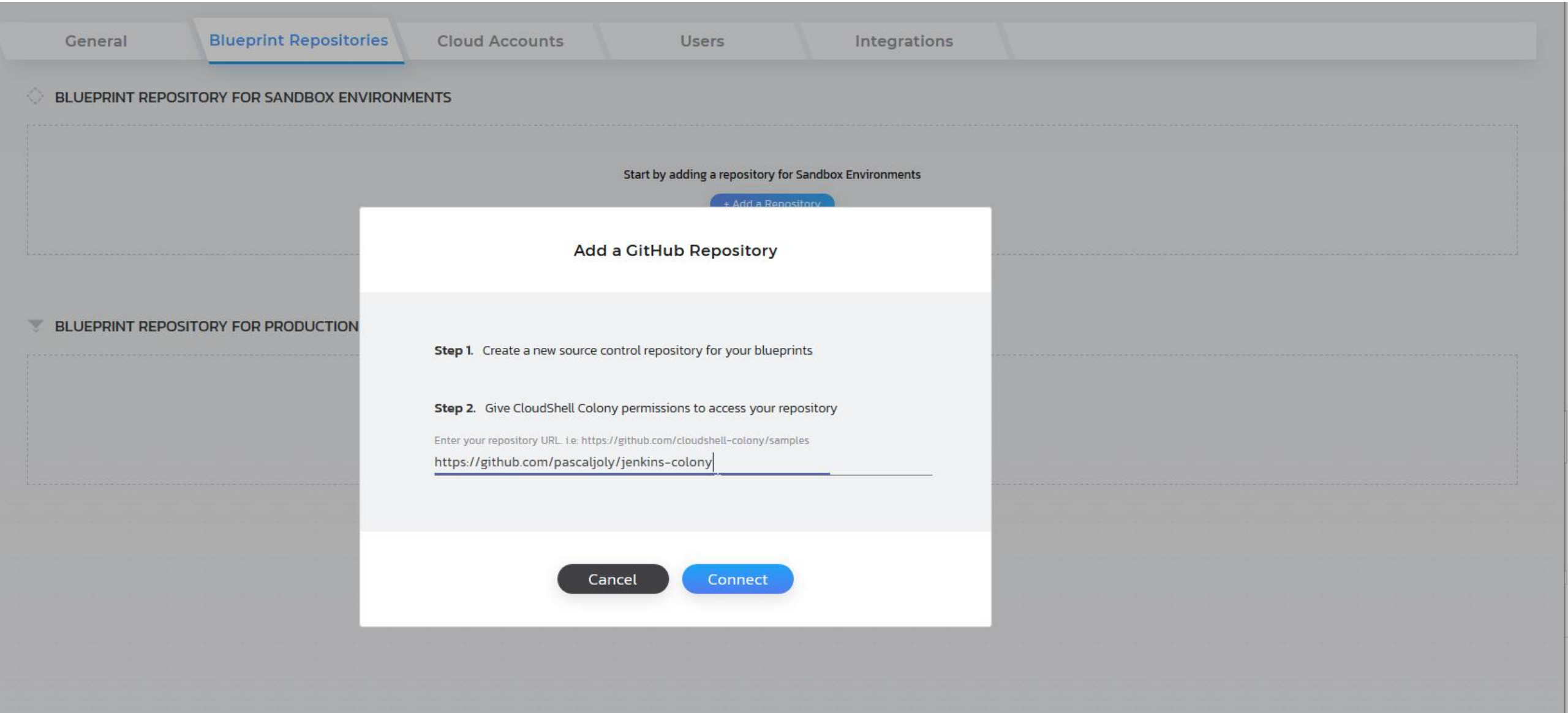

Edit Cloud Account
GENERAL
AWS
SPACES
Select space
Add
Trial
Ops-Tools
Internet facing
Use shared Virtual Network
Cancel
Save


Cloud Accounts
+ Add Cloud Account
Search by name
ACCOUNT
CLOUD PROVIDER ID
AUTHENTICATED
AUTHENTICATED BY
USED IN SPACE
COMPUTE SERVICES
AWS
754261393455
Oct 1, 2020 11:14 PM
PJ
pascal joly
Trial, Ops-Tools

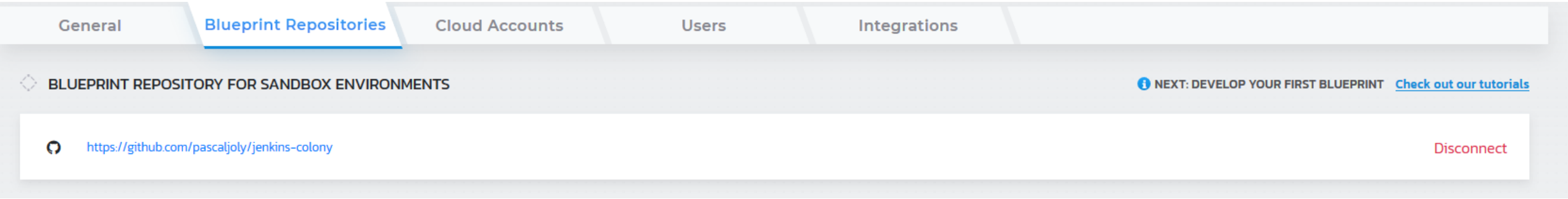
Add blueprint repository to the Ops-Tools space

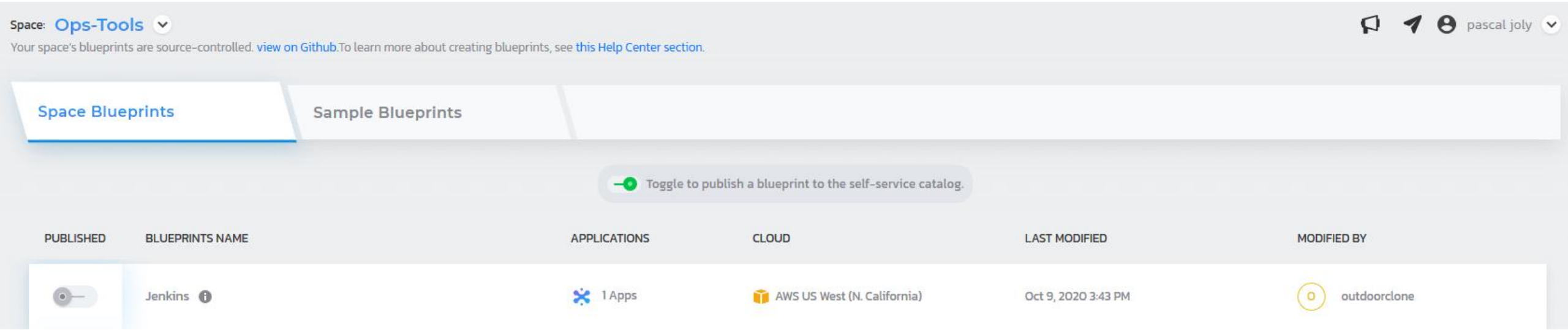


Use the Jenkins repo your forked earlier in Git

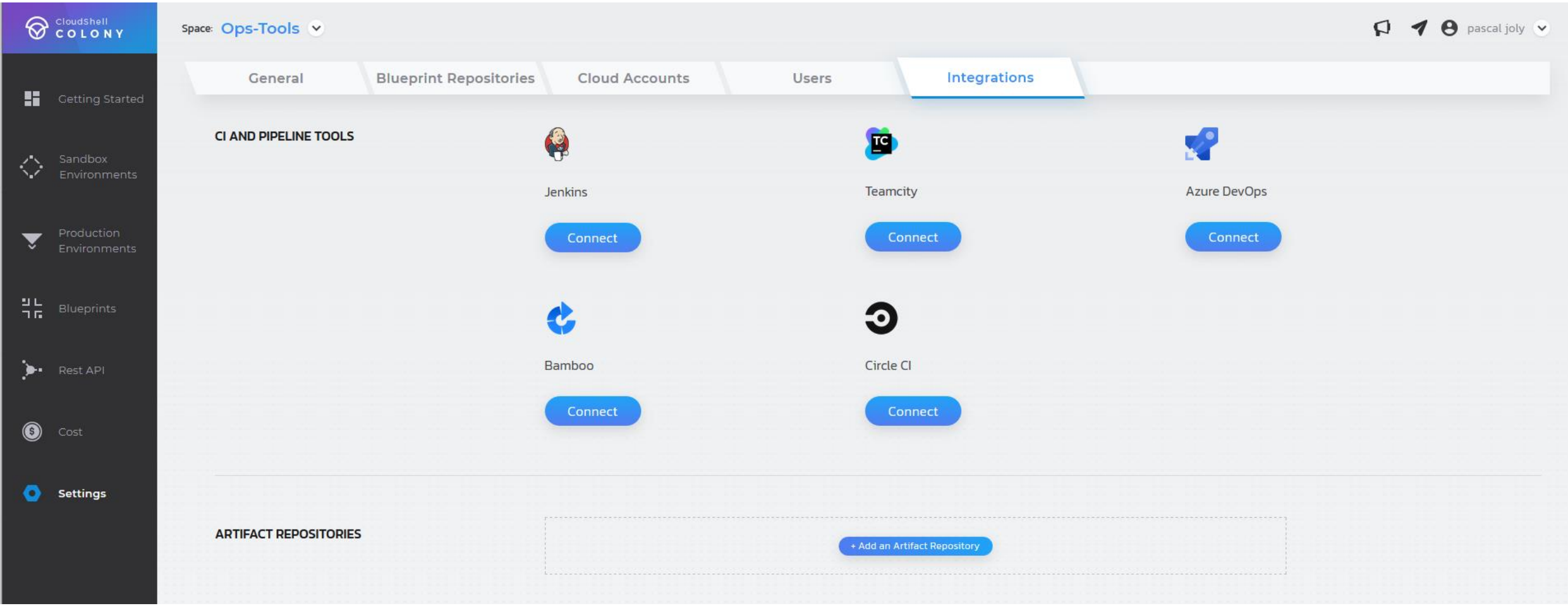


The Jenkins blueprint is now listed in your blueprint catalog

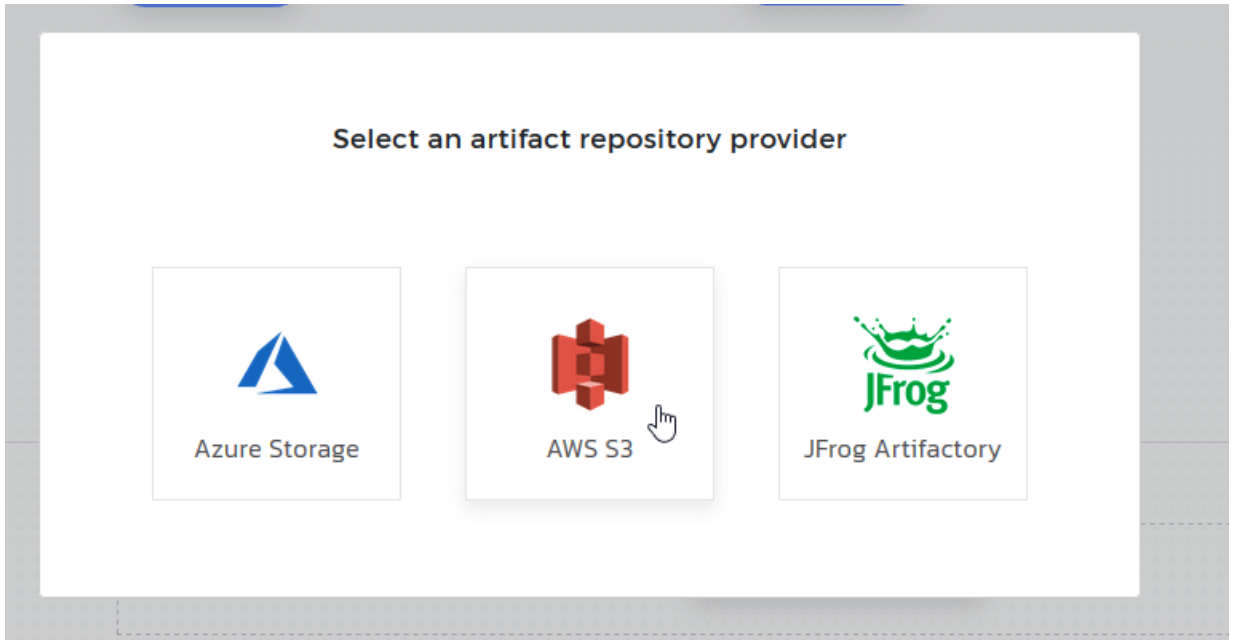




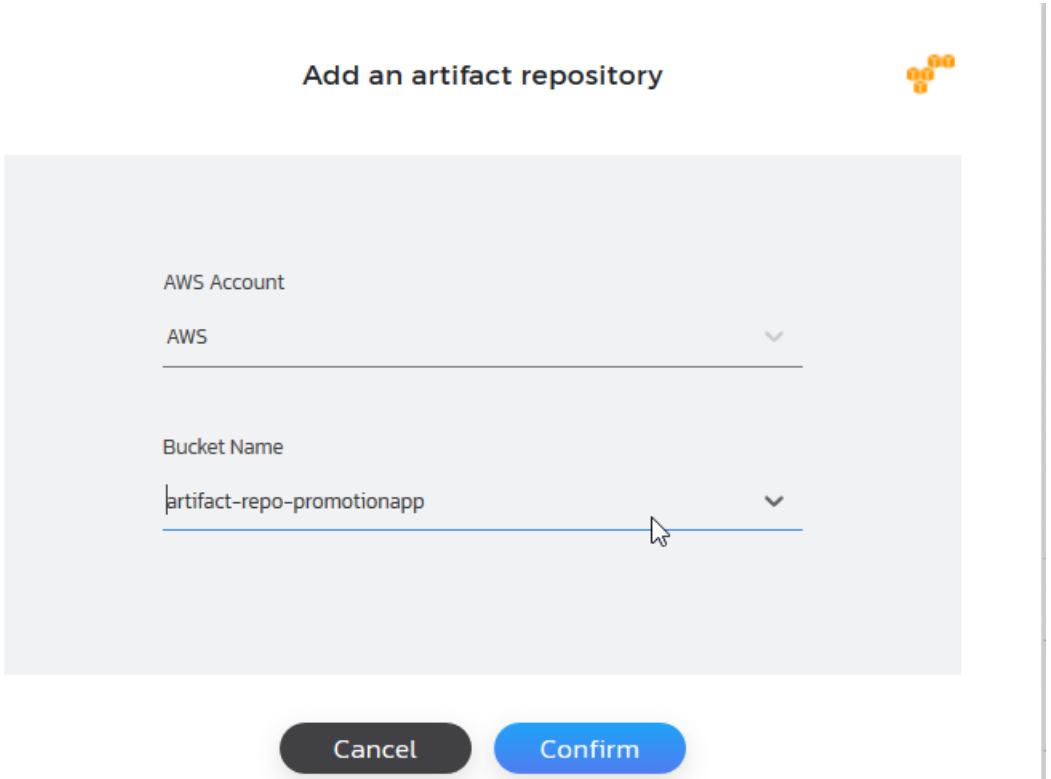
Go to Settings and add the artifact repository (S3 bucket)

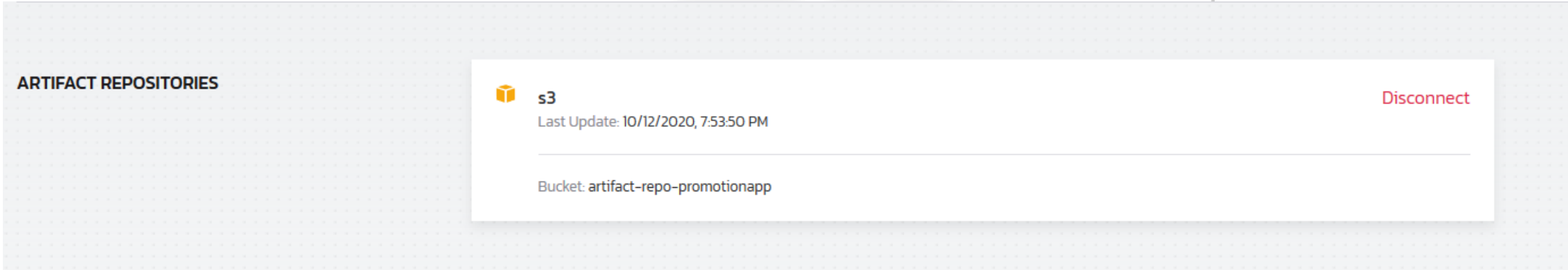


Select AWS S3

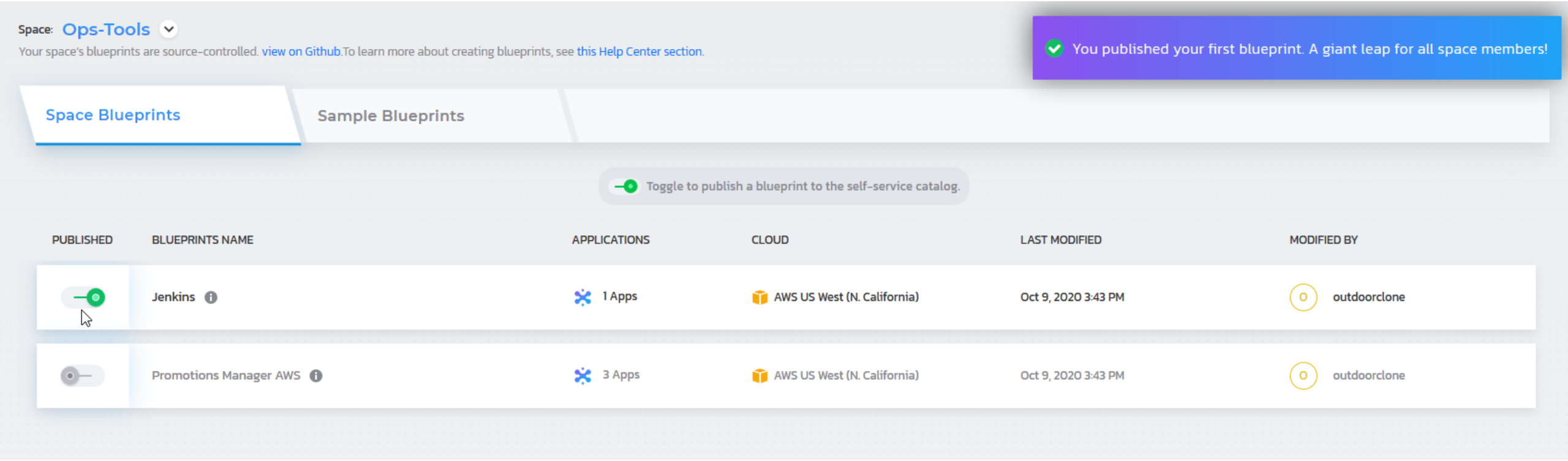


Select from the pull down the cloud provider (AWS) and the S3 repo you previously created in Module 1

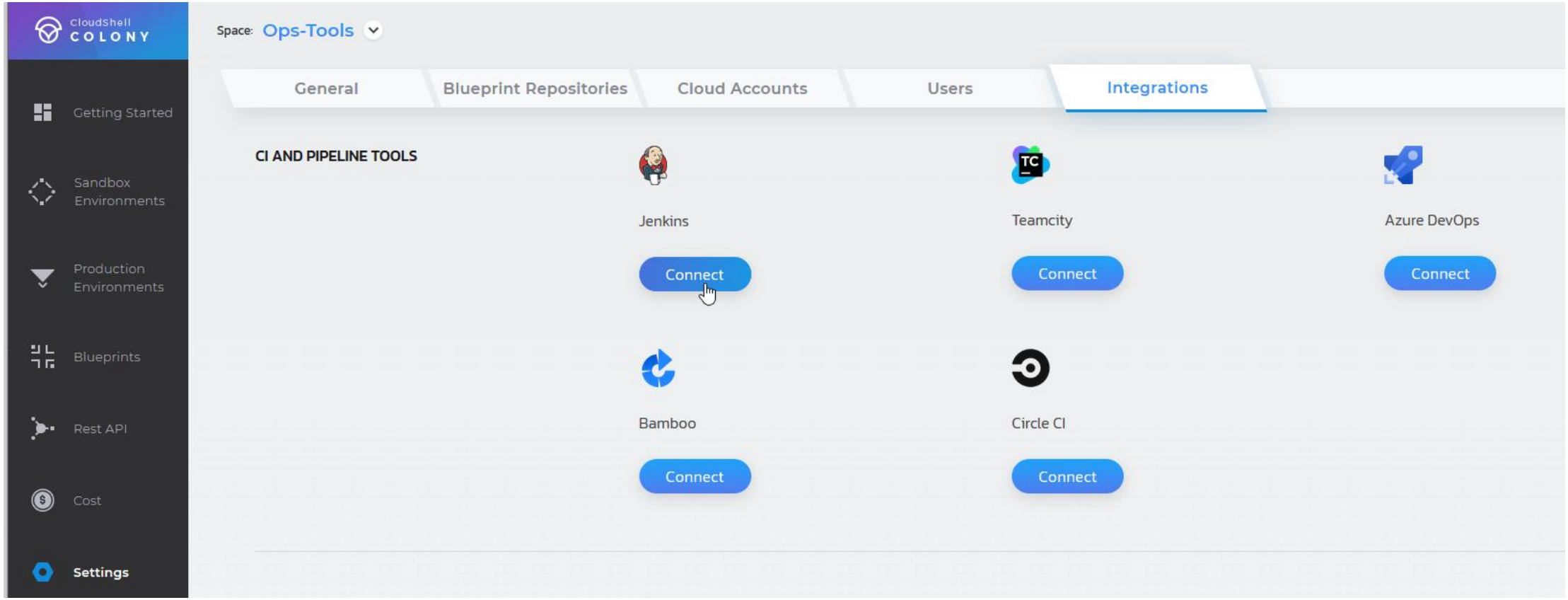




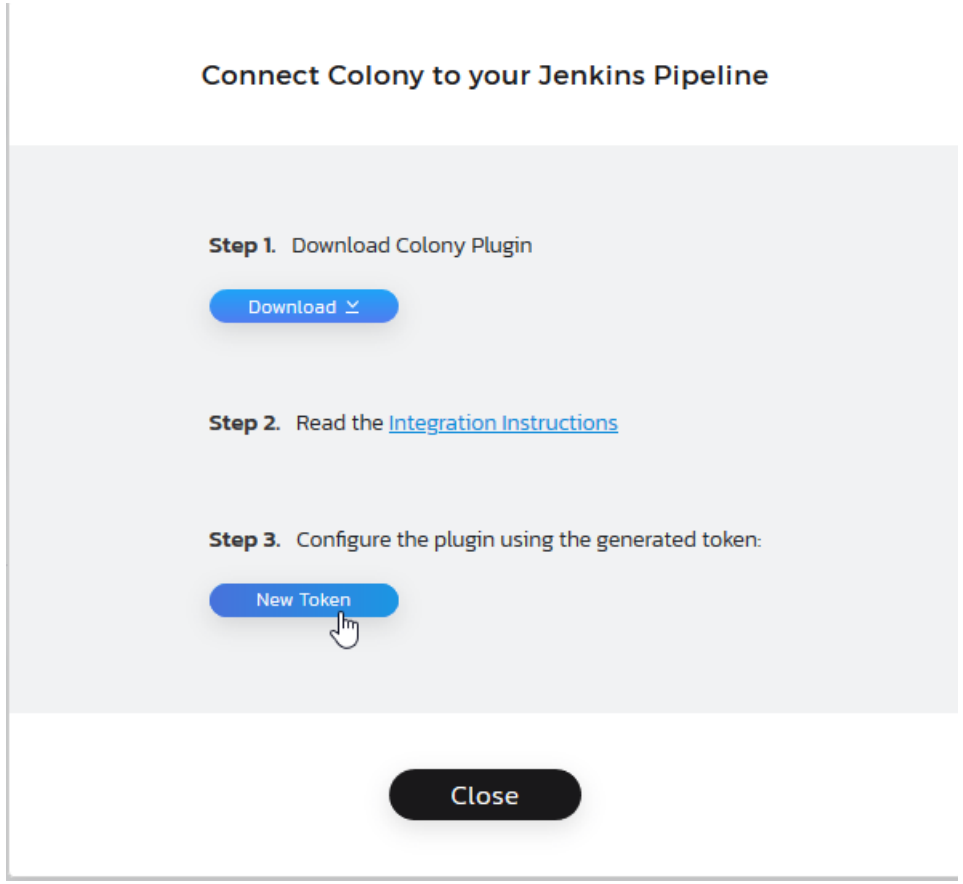
Publish Jenkins Blueprint: toggle the slider



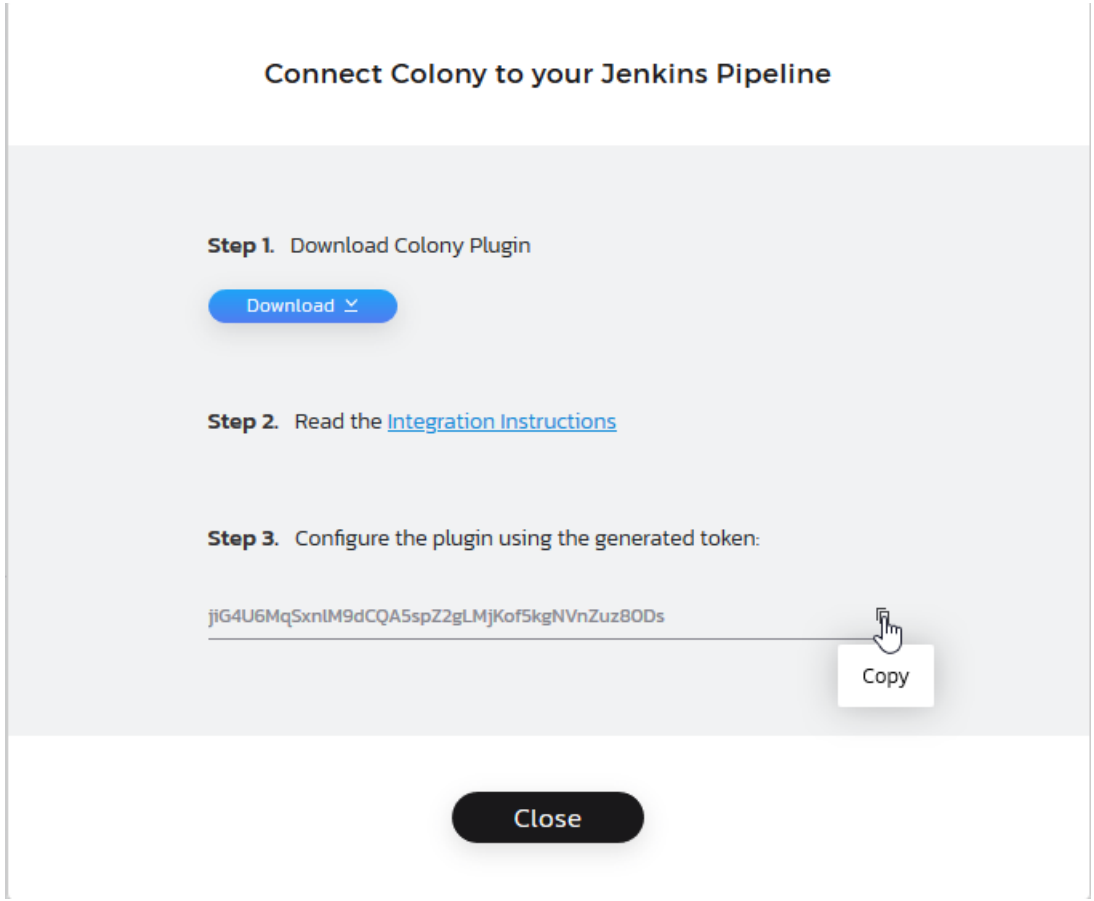
Navigate to Settings > Integrations and click Connect under the Jenkins icon



Generate Colony token for Jenkins: click on New Token

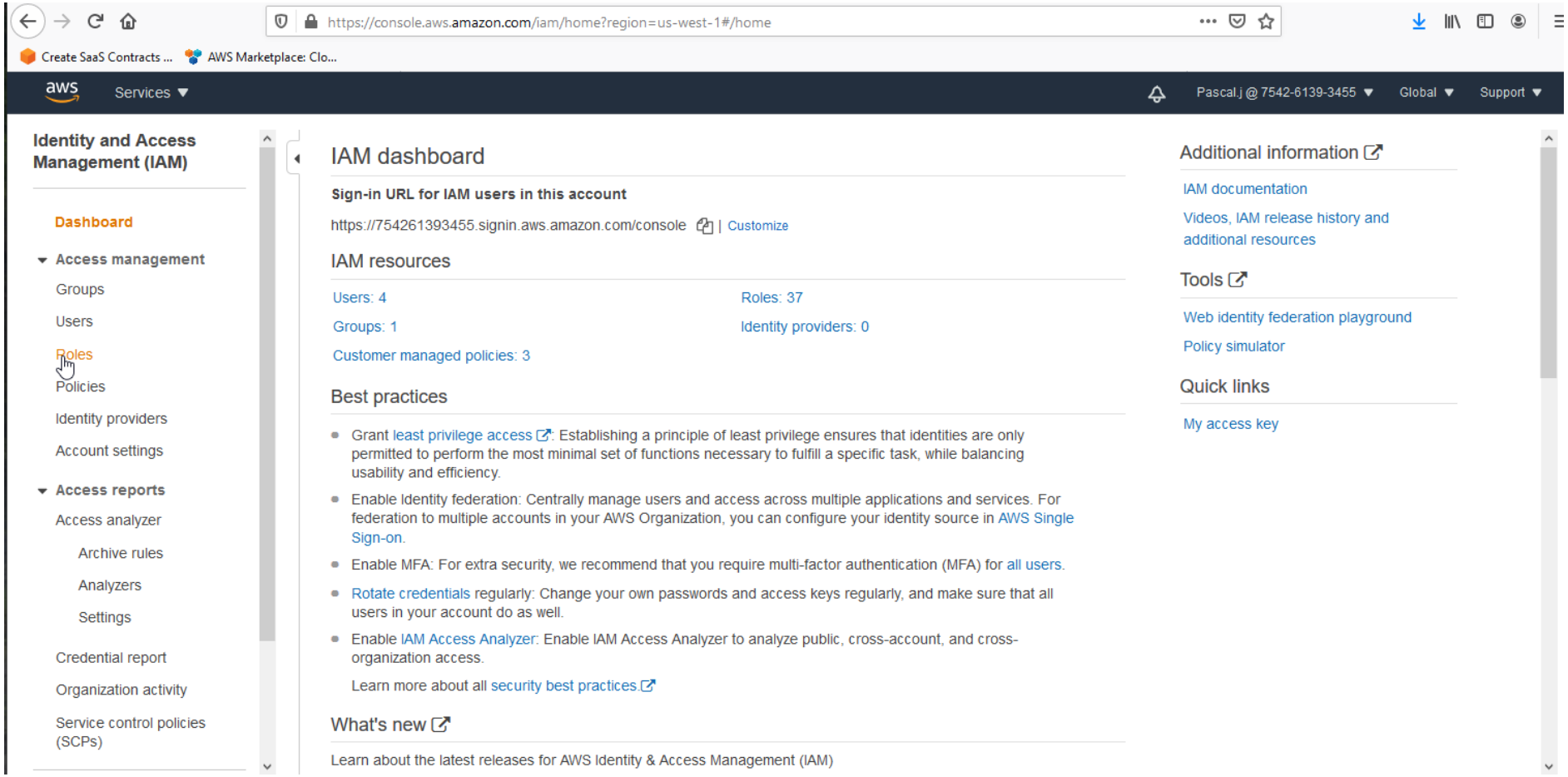


Save the token in a notepad for future use: this will be an input parameter of your sandbox

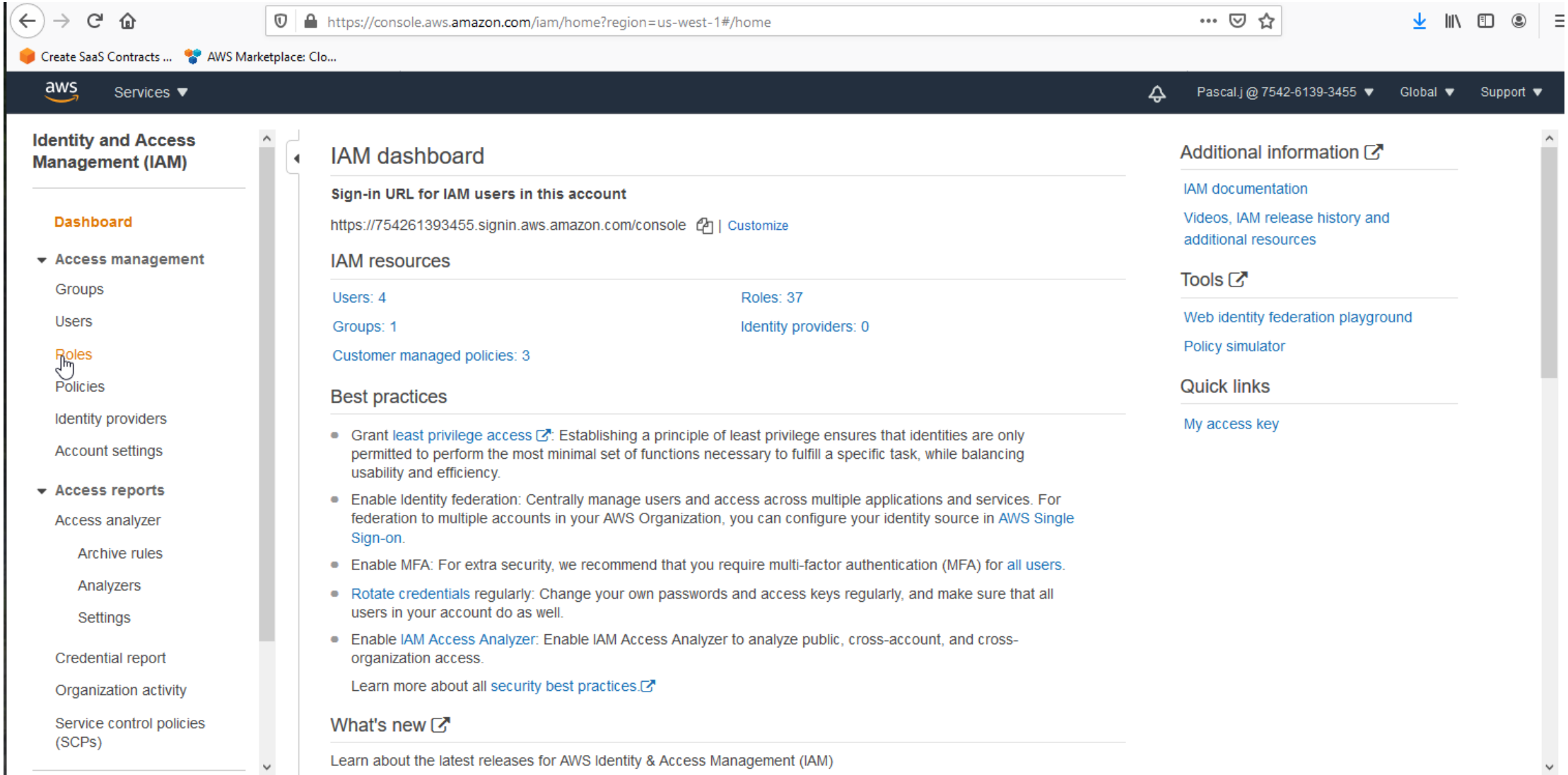


In AWS Console, create a IAM role with write access to the S3 bucket

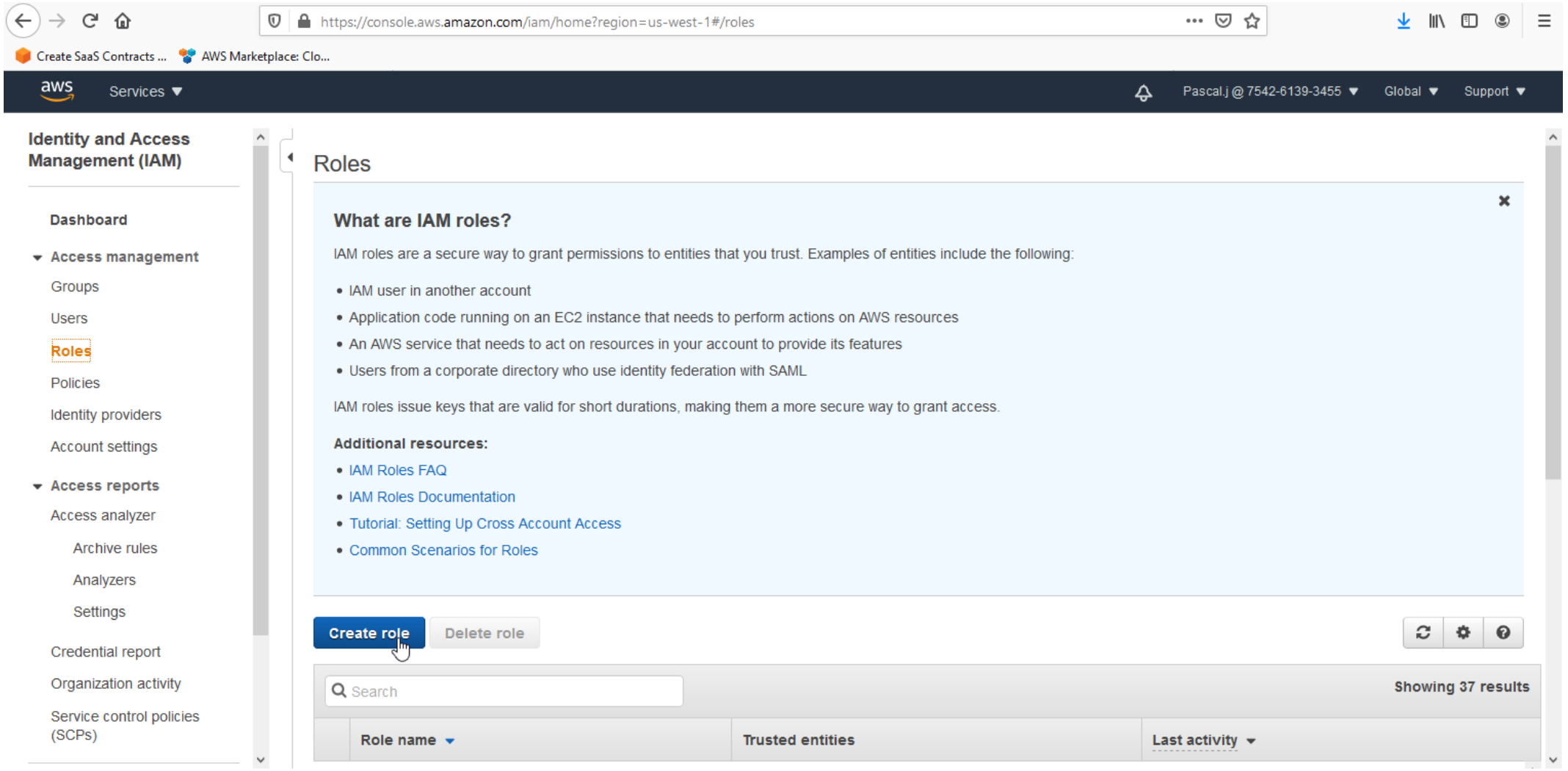
Navigate to Services > IAM



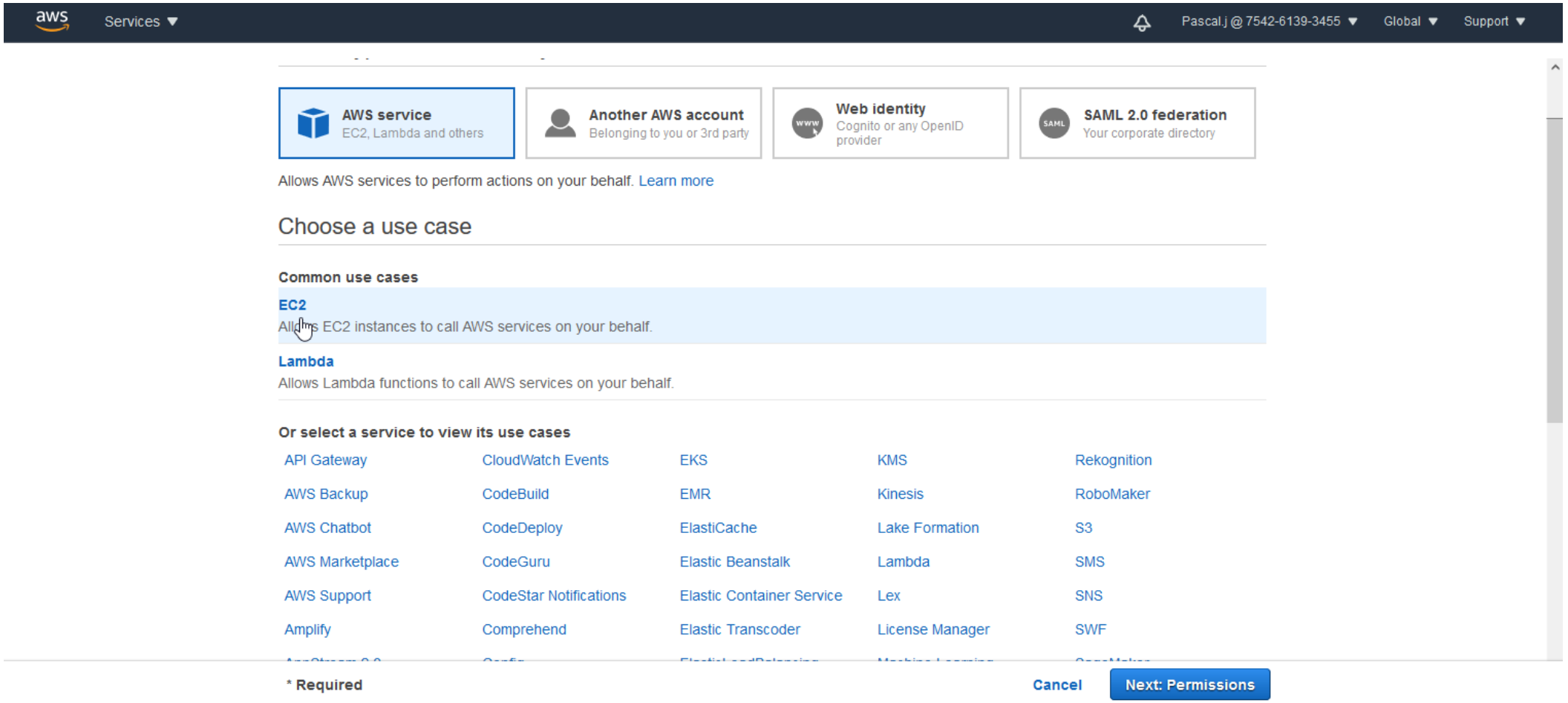
Navigate to the IAM service dashboard and click on Roles



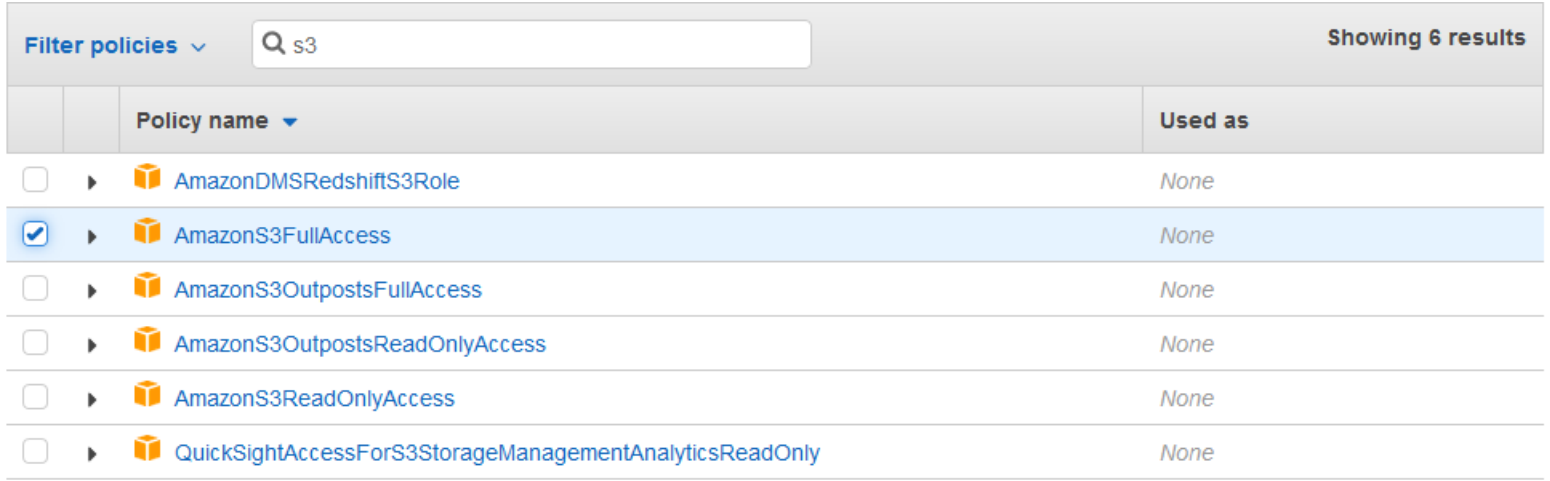
Create Role


https://console.aws.amazon.com/iam/home?region=us-west-1#/roles
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

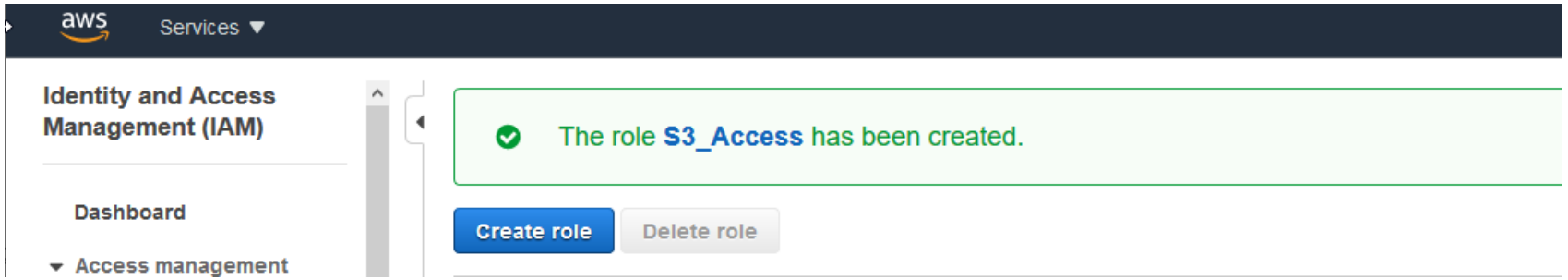


Select AmazonS3FullAccess Policy



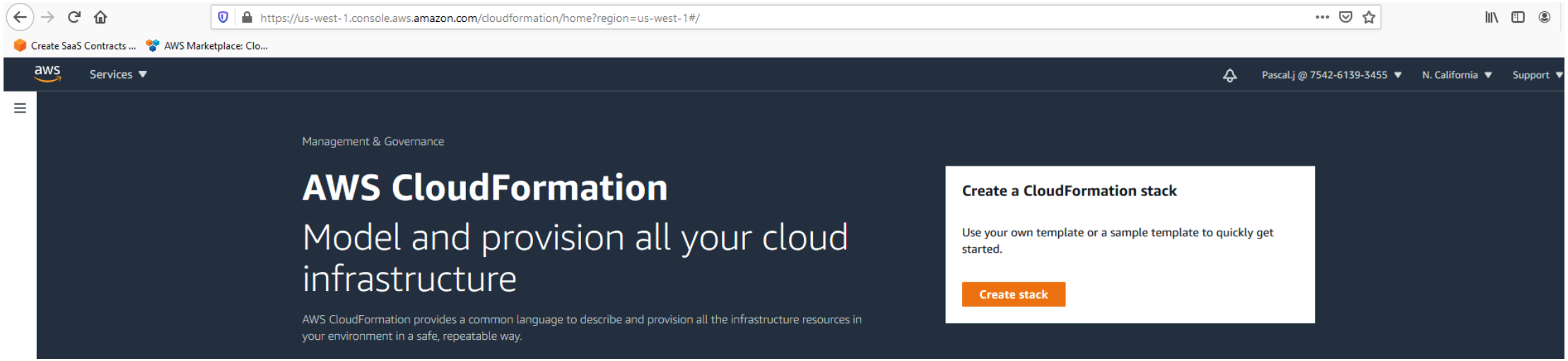
Name your Role S3_Access and Create role

This role will be an input of your sandbox

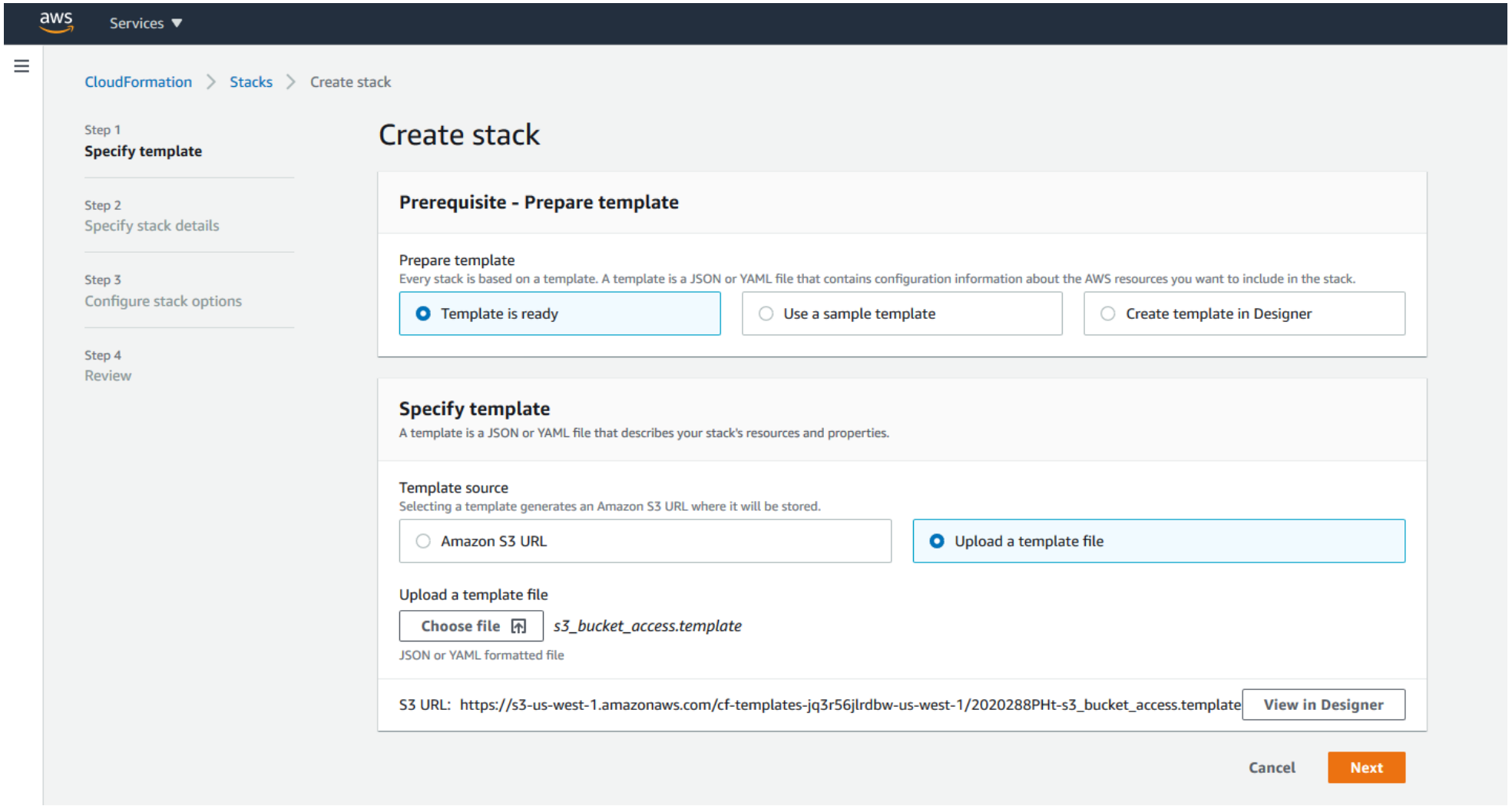


generate the S3 role using a CF template

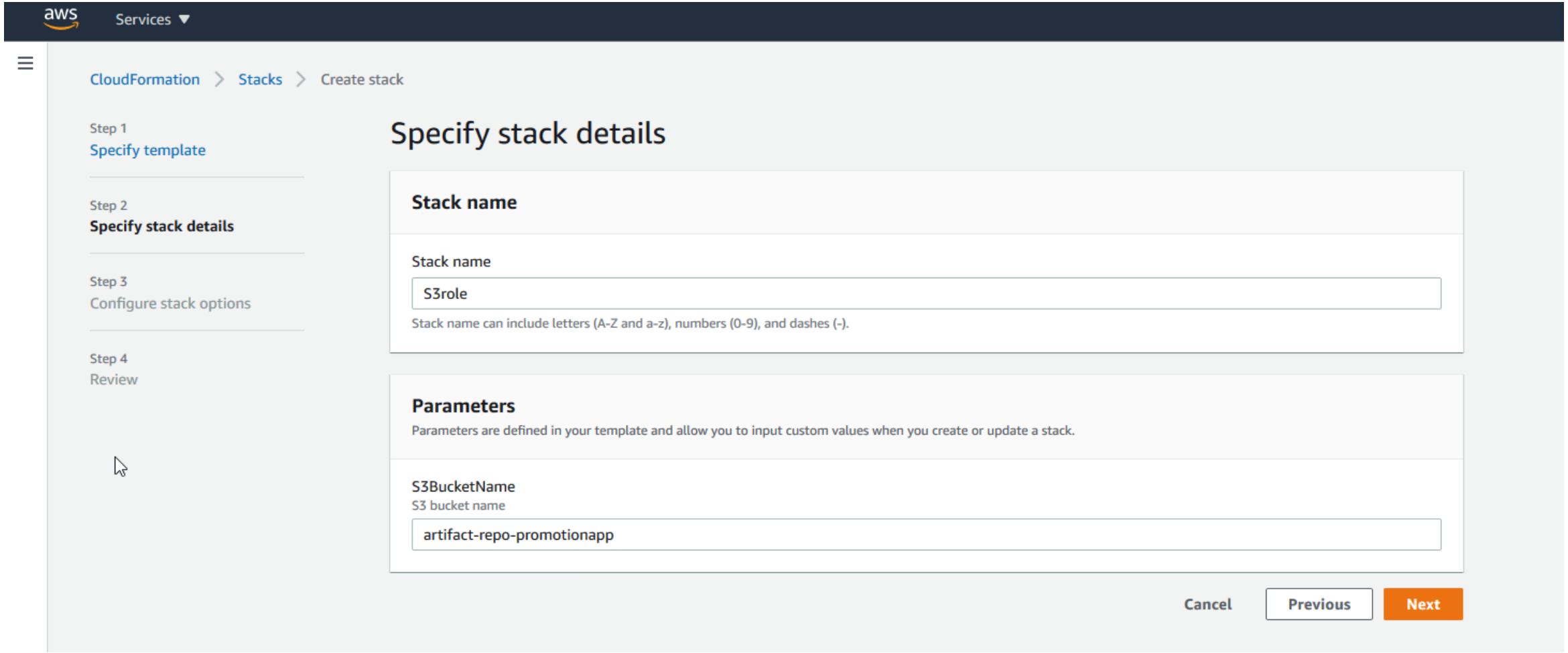
Navigate to Services: Cloud Formation to Create Stack



Upload template file *s3_bucket_access.template* provided in the workshop

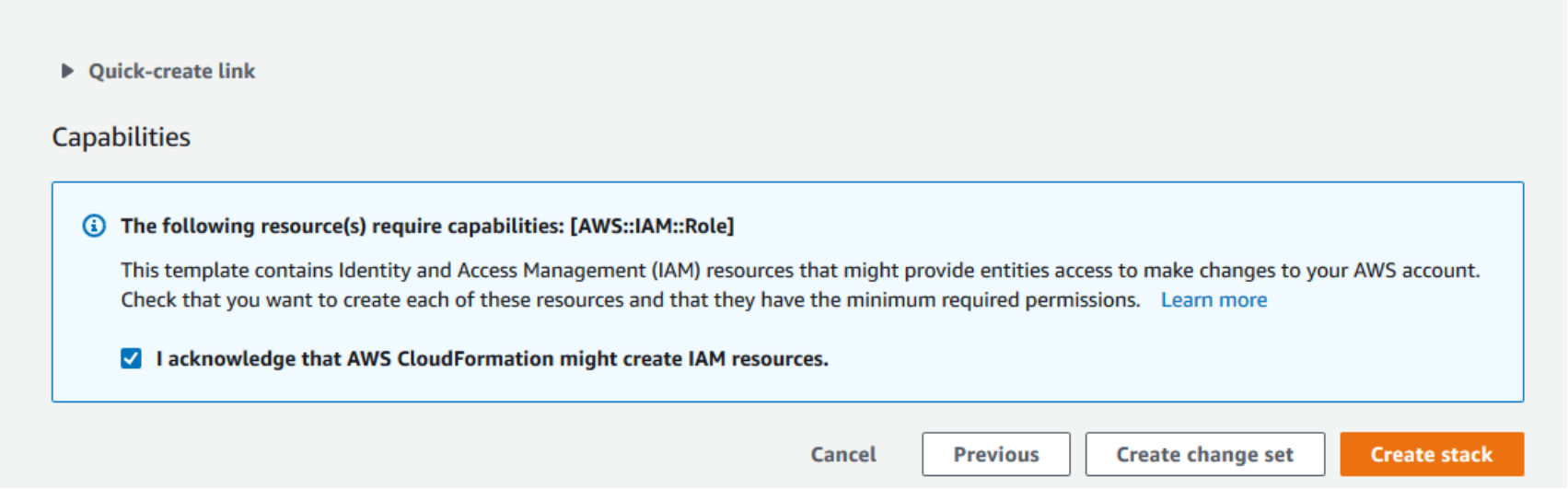


Enter a name for the stack and specify the S3 bucket name as input parameter

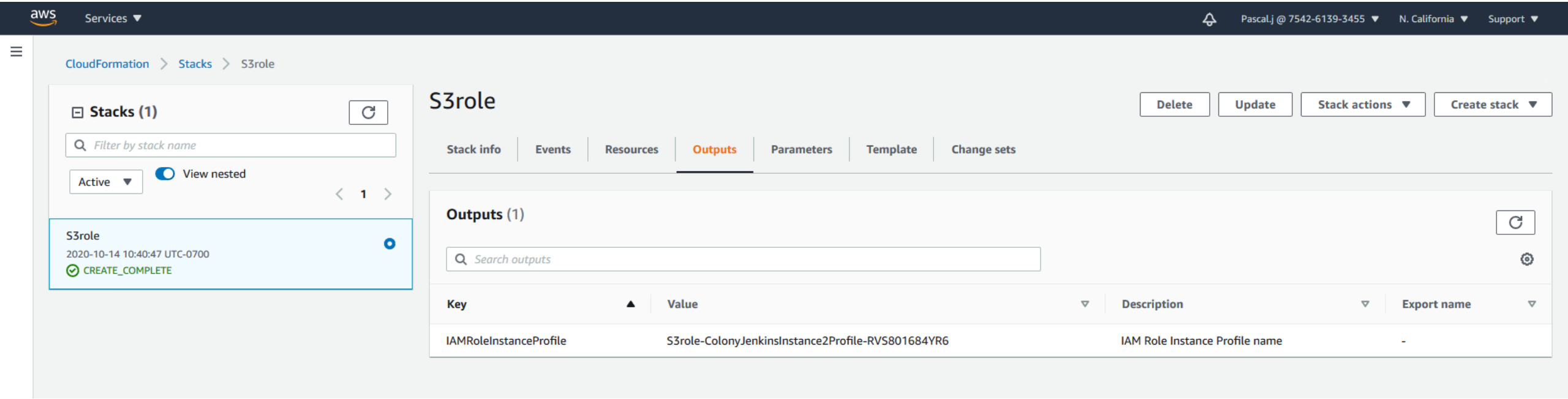


Follow the Next screens until you get to the stack creation button (check box at the bottom)

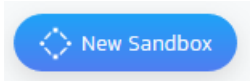
All other values can be kept default

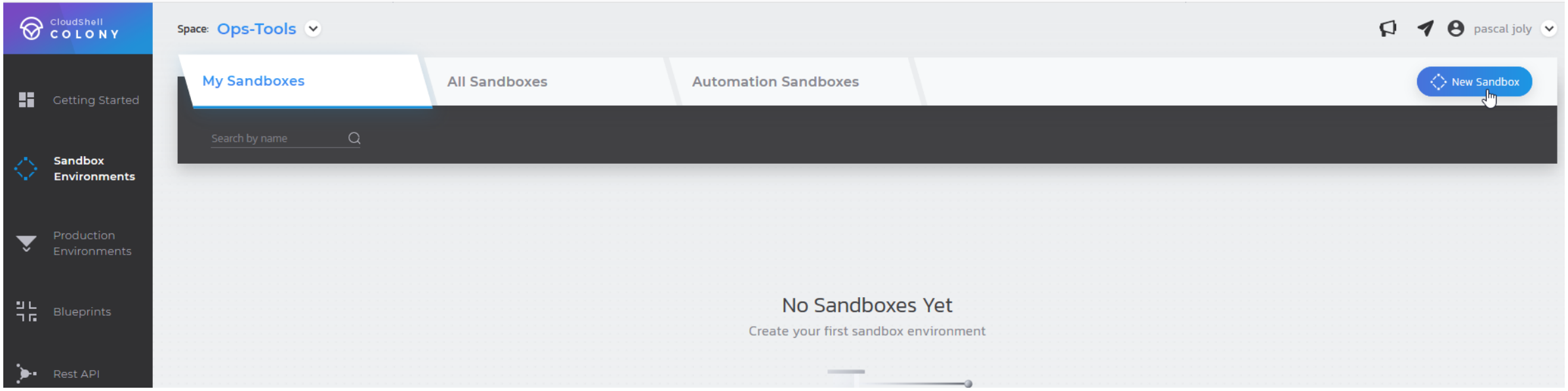


S3 role is now created. This role will be an input of your Jenkins sandbox

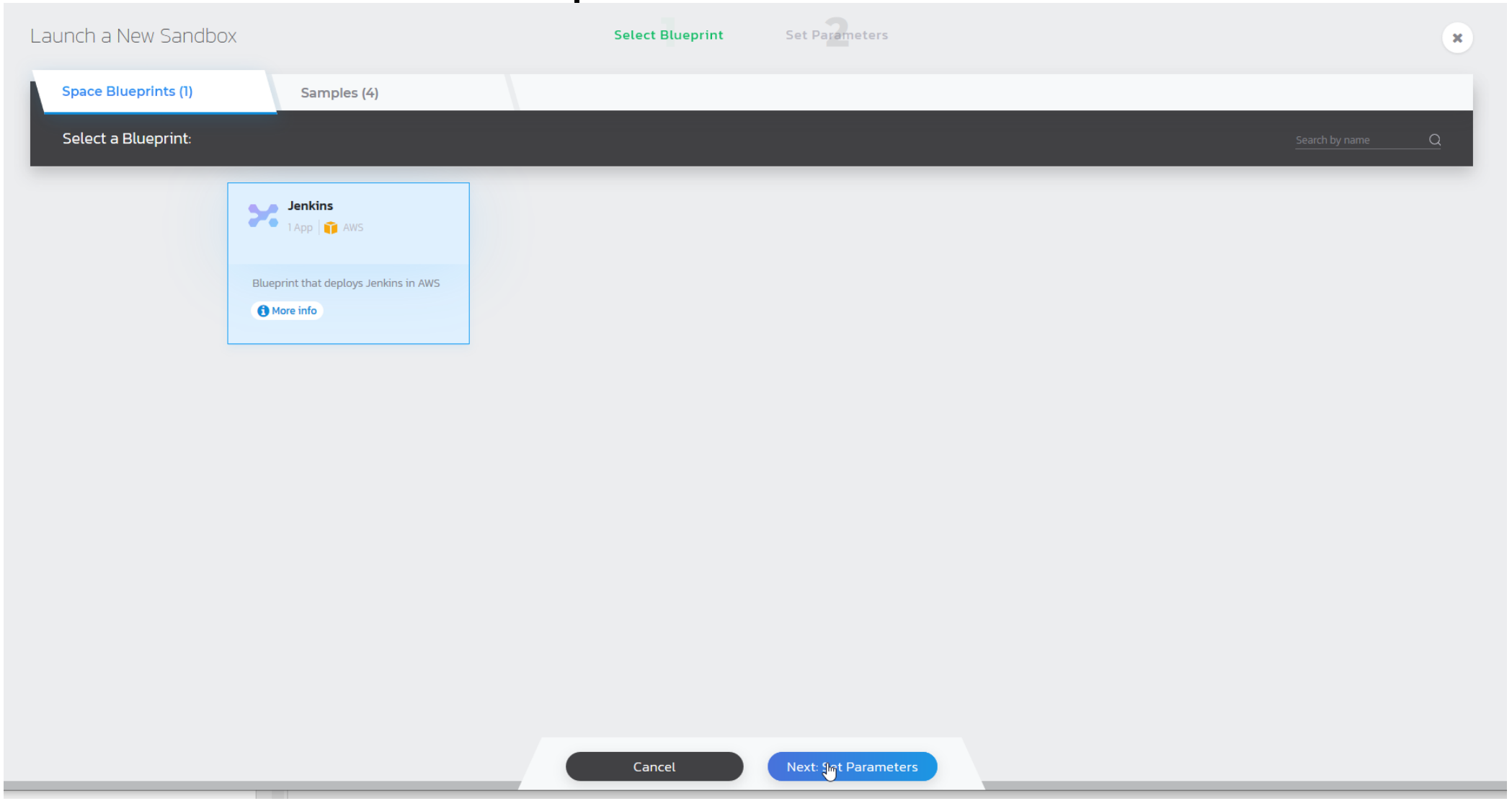


Deploy Jenkins Sandbox in Colony

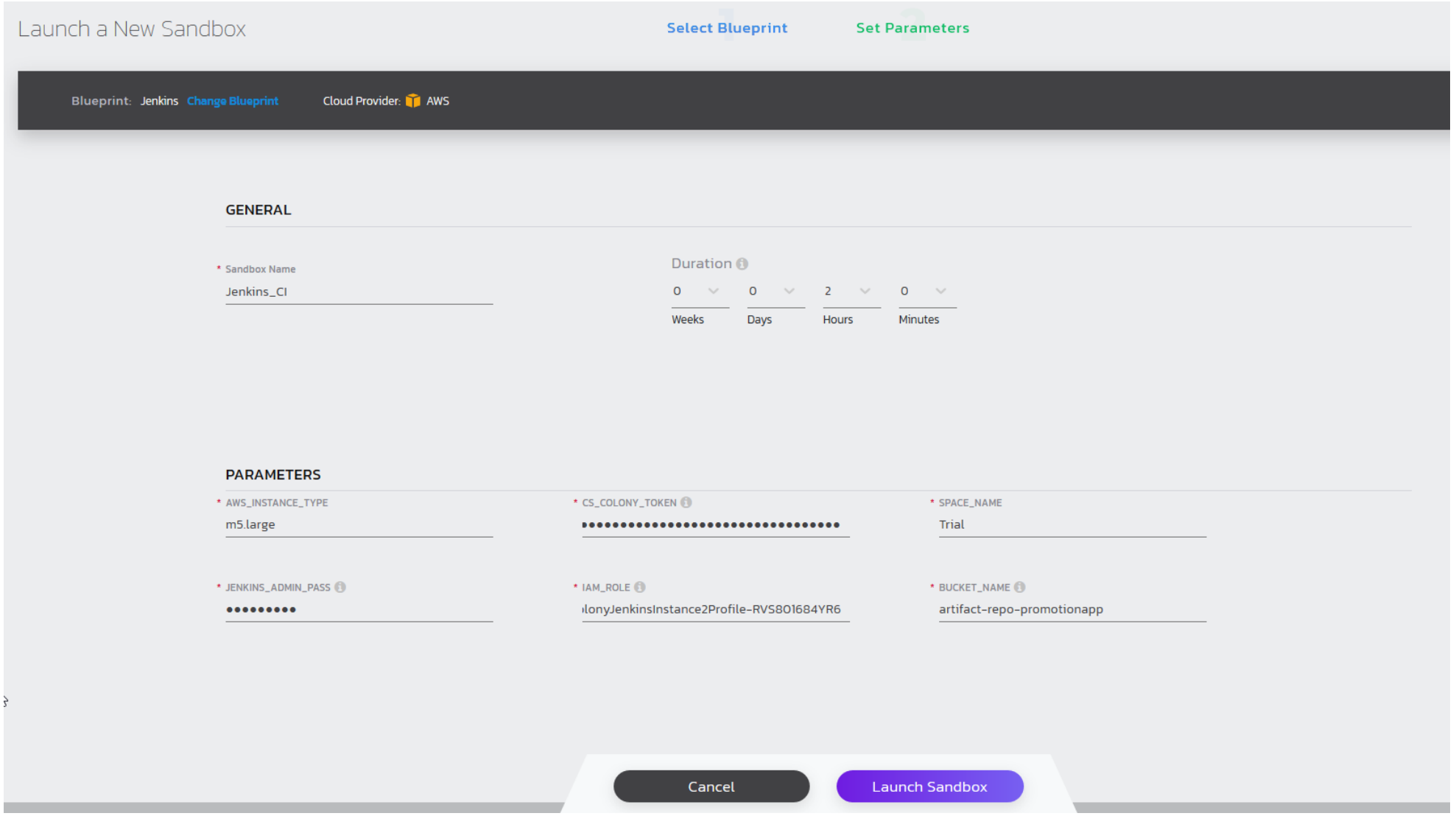
Navigate to the Sandbox Environment of the OpsTools space in Colony.
Click on 



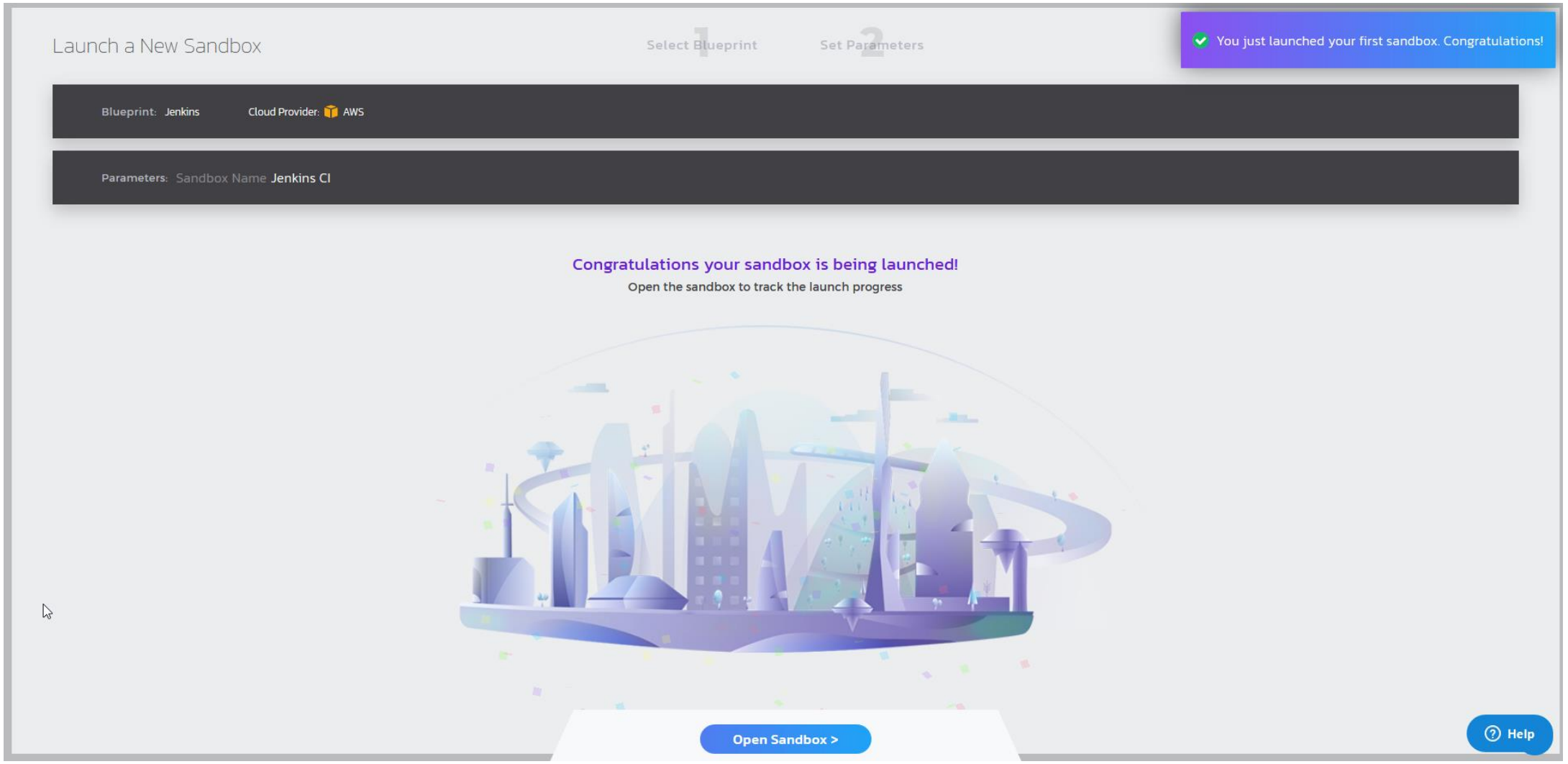
Select Jenkins blueprint and click Next



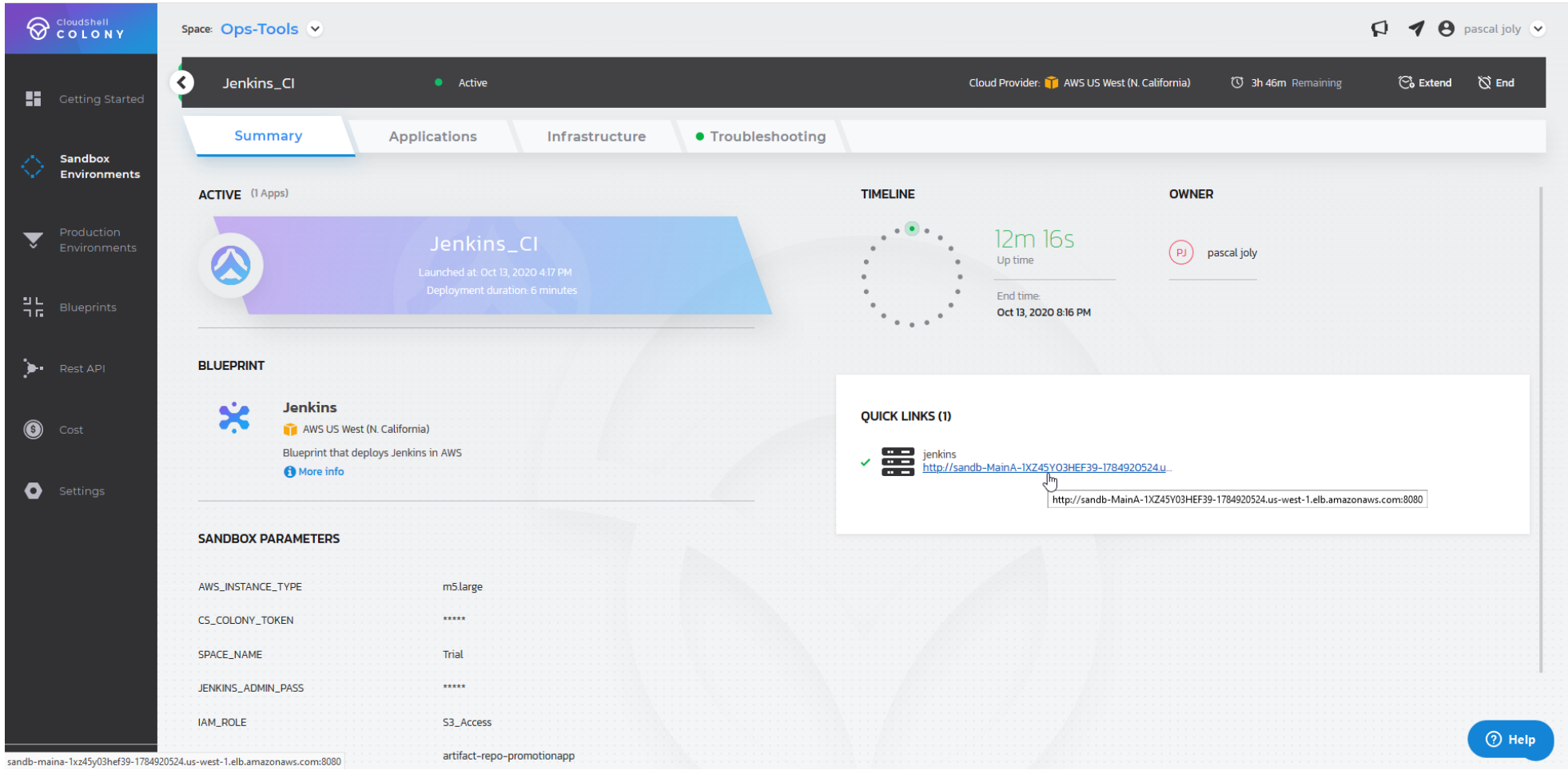
Enter required parameters as below



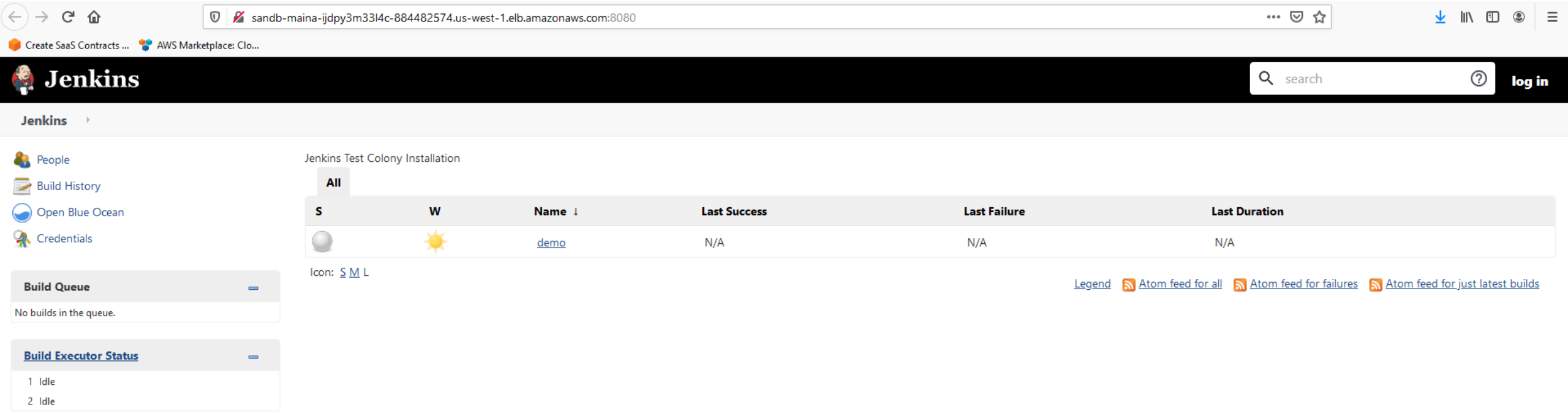
Launch Sandbox



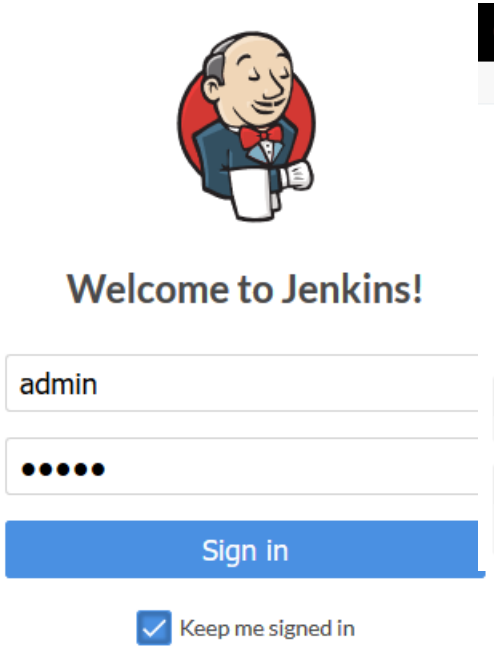
Once the sandbox deployment is complete, click on the web “quick link” to access Jenkins

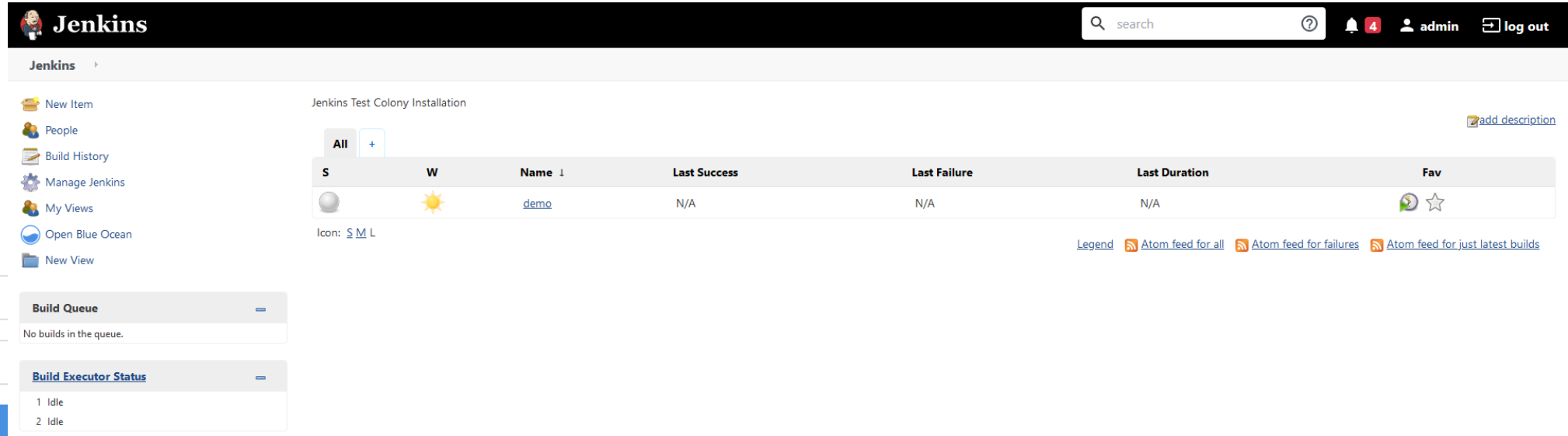


Login to the console



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





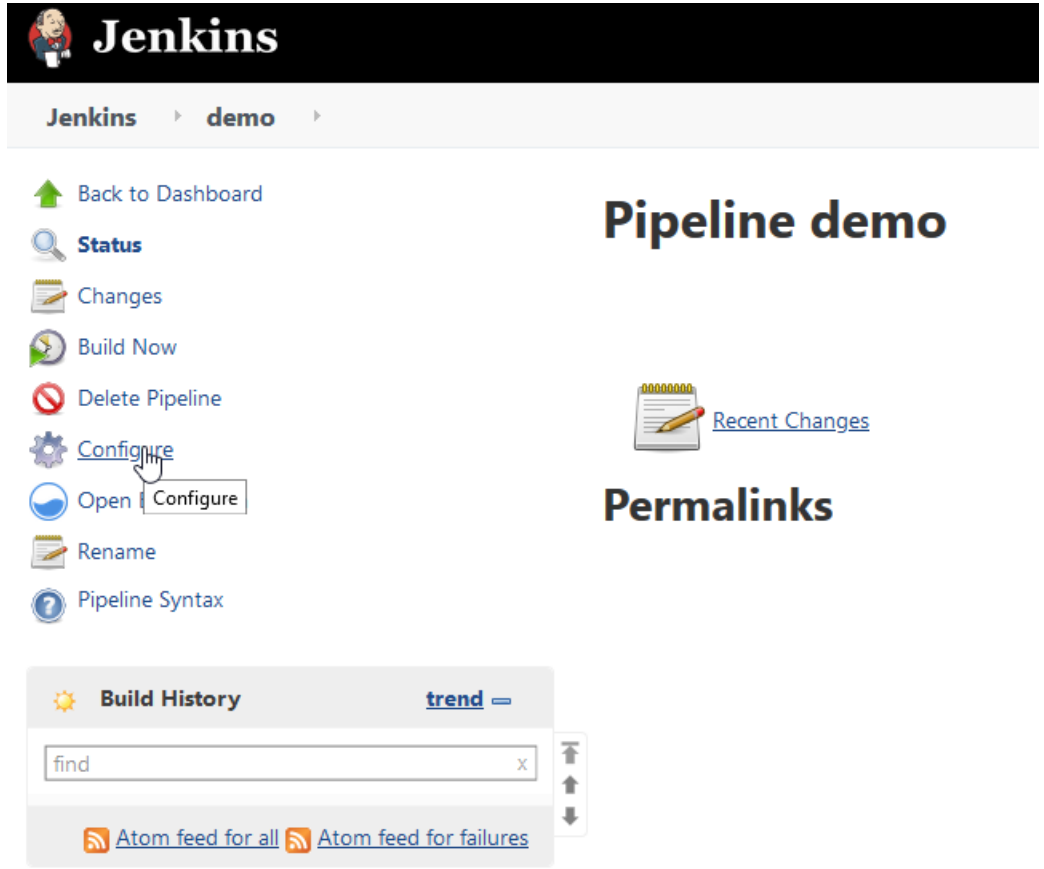
Deploy a sandbox from Jenkins.

- 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

Click Configure



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-sandobx-' + 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-sandobx-' + 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

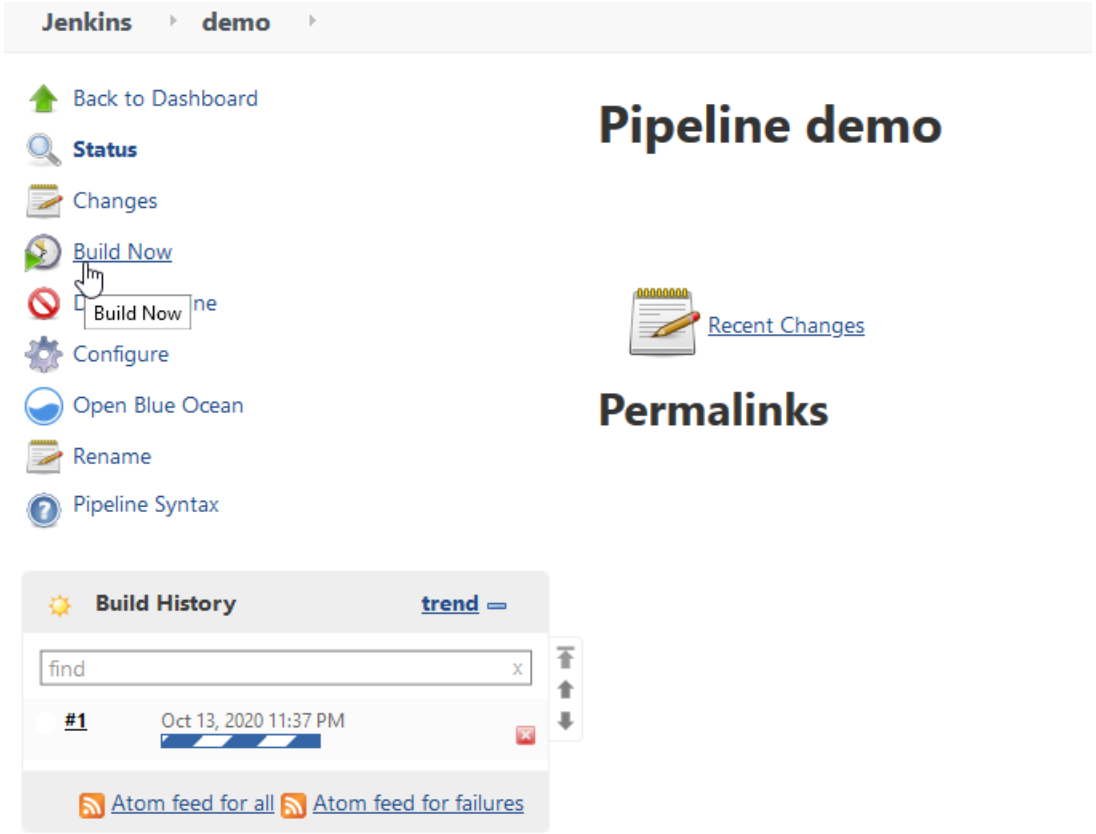
Pipeline Syntax

Pipeline steps

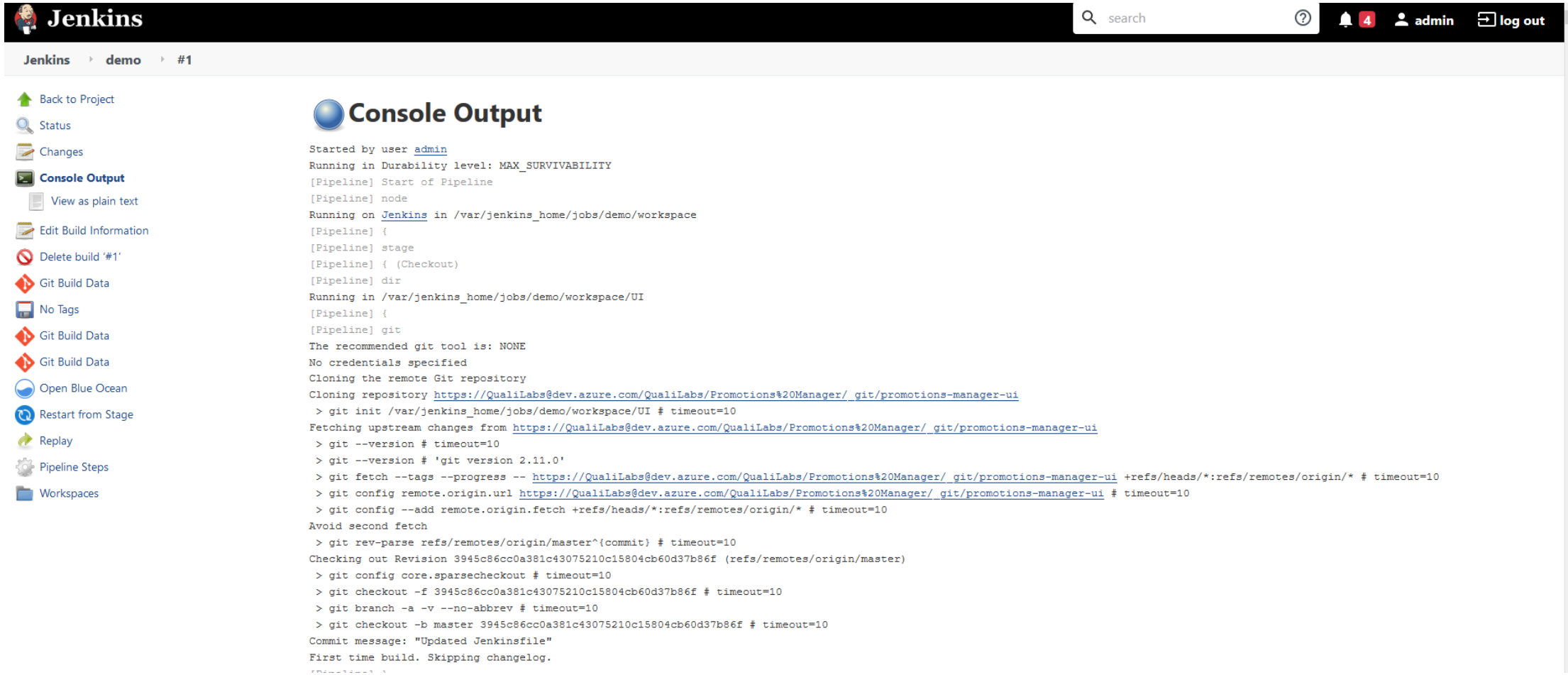
The default pipeline loaded with this Jenkins instance will automatically run the following steps:

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



Check results/console Output

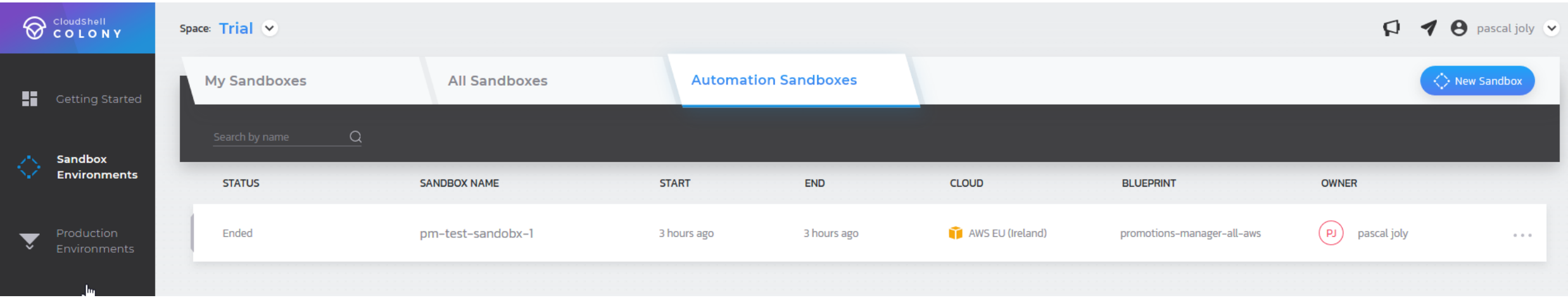


Check the sandbox deployment in the output

```
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



Learn More

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