Deploy a custom app with Terraform

Jennifer Hilburn

Okta Education Services

8.2021

okta

Safe harbor

This presentation contains "forward-looking statements" within the meaning of the "safe harbor" provisions of the Private Securities Litigation Reform Act of 1995, including but not limited to, statements regarding our financial outlook, product development, business strategy and plans and market trends, opportunities and positioning. These forward-looking statements are based on current expectations, estimates, forecasts and projections. Words such as "expect," "anticipate," "should," "believe," "hope," "target," "project," "goals," "estimate," "potential," "predict," "may," "will," "might," "could," "intend," "shall" and variations of these terms and similar expressions are intended to identify these forward-looking statements, although not all forward-looking statements contain these identifying words. Forward-looking statements are subject to a number of risks and uncertainties, many of which involve factors or circumstances that are beyond our control. For example, the market for our products may develop more slowly than expected or than it has in the past; our operating results may fluctuate more than expected; there may be significant fluctuations in our results of operations and cash flows related to our revenue recognition or otherwise; a network or data security incident that allows unauthorized access to our network or data or our customers' data could damage our reputation; we could experience interruptions or performance problems associated with our technology, including a service outage; we may not be able to pay off our convertible senior notes when due; and global economic conditions could deteriorate. Further information on potential factors that could affect our financial results is included in our most recent Annual Report on Form 10-K and our other filings with the Securities and Exchange Commission. The forward-looking statements included in this presentation represent our views only as of the date of this presentation and we assume no obligation and do not intend to update these forward-looking statements.

Any unreleased products, features or functionality referenced in this presentation are not currently available and may not be delivered on time or at all. Product roadmaps do not represent a commitment, obligation or promise to deliver any product, feature or functionality, and you should not rely on them to make your purchase decisions.

Table of contents

01

Introduction

- Okta Ice
- Deploying the SPA sample code
- Lab Recipe

02

The Okta and ASA Terraform Providers

Automate your DevOps needs

03

Download the Terraform Configs

Everything you need

04

Run the Configs

- Configure your SPA app in your org
- Create a new ASA project and
- Deploy your AWS server

05

Test your Deployed App Server

- Login as a test user on the web
- Access server with SSH

06

Destroy the Test Environment

Rinse and Repeat!

The team

Jennifer Hilburn

Senior Technical Instructor

okta Certified Professional

okta Certified Administrator

okta Certified Instructor

Eric Lynch

Senior Technical Instructor

okta Certified Developer

Dave Silvestro

Senior Curriculum Developer

okta Certified Professional

okta Certified Administrator

okta

01

Introduction

- Okta Ice
- Deploying the SPA sample code
- Lab Recipe

Okta Ice

Okta Ice Gourmet Ice Cream

CHALLENGE

On premises Apps with legacy authentication

Implement Agile

Insecure access to test environment

SOLUTION

Modern Apps

Okta IDaaS

Okta ASA

Terraform

> “It is not the strongest of the species that survive, nor the most intelligent, but the one most responsive to change”

SARAH JAMES, OKTA ICE CIO

Deploying your App

DEVELOP

- Vue Single Page App
- Okta Vue SDK
- Okta Auth JavaScript SDK
- Okta Authentication API
- Okta Developer Org
- Node.js server
- http://localhost:8080/
- Access: Developer only

TEST

- Vue SPA production-ready build
- Okta Sandbox Org
- Ubuntu Linux AWS Instance
- ASA Server Agent
- Apache 2 http server
- https://*awsinstance*
- Access: Developer and Test team

Never Deploy on Friday

Lab Recipe

Develop

- Node.js
- Okta Developer Org
- Okta Vue Sample Application
  - https://github.com/okta/samples-js-vue
- Github repo (optional)

Prepare

- Okta Sandbox Org
  - Users, Groups, API token
- Okta ASA Team
  - Users, Groups, Service Account
- AWS account
- ASA Client
- Terraform host with binary
- Github client to access repo

Deploy

- Okta Dev Day DevOps Terraform configs
  - https://github.com/OktaEdu/es-dev-day-21

02

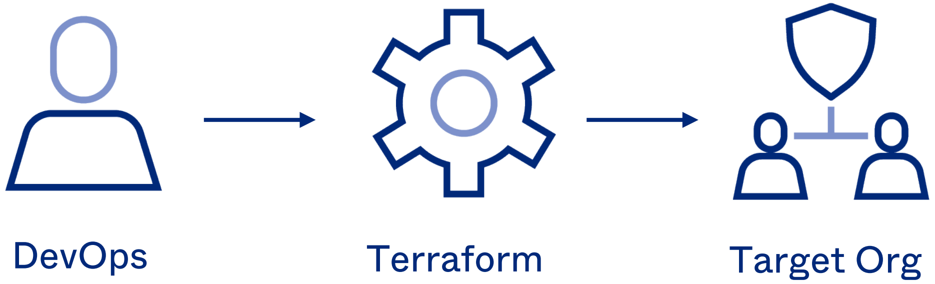
The Okta and ASA Terraform Providers

Automate your DevOps needs

okta

Okta Terraform Provider

- https://registry.terraform.io/providers/okta/okta
- https://github.com/okta/terraform-provider-okta
- Required variables (environment or provider config)
  - org_name
  - base_url
  - api_token
- Monitor rate limits
- Leverages and provides similar functionality to API
- Go version 1.1.5+



Okta ASA Terraform Provider

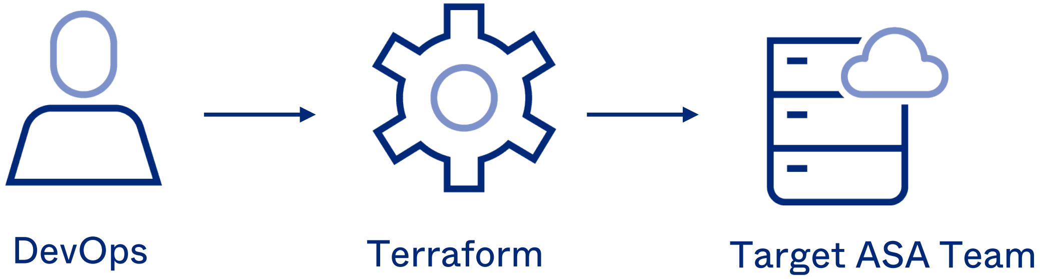
- https://registry.terraform.io/providers/oktadeveloper/oktaasa

- https://github.com/oktadev/terraform-provider-oktaasa

- Required environment variables
  - OKTAASA_KEY
  - OKTAASA_KEY_SECRET
  - OKTAASA_TEAM

- Manage:
  - Projects
  - Groups
  - Enrollment Tokens

- Go version 1.1.3+



okta

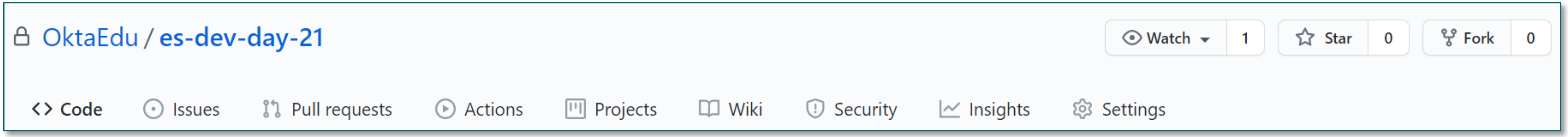
03

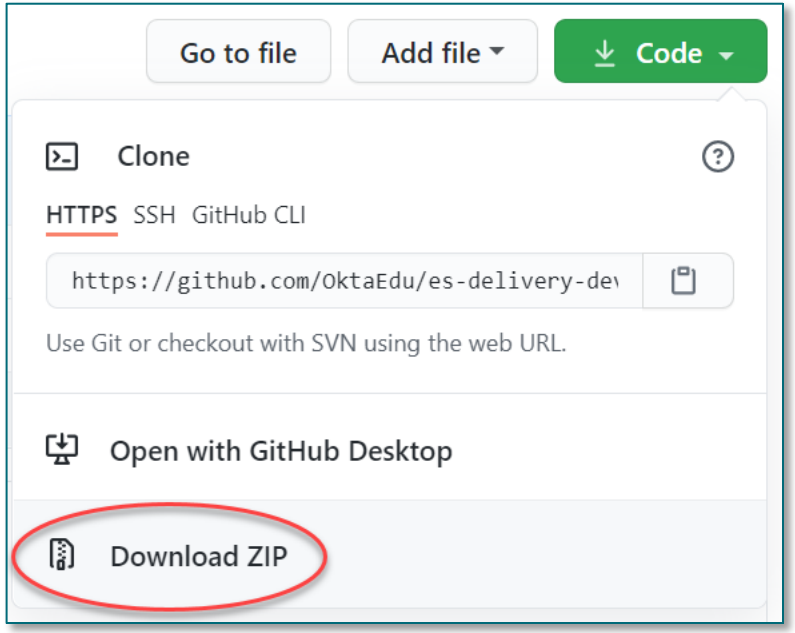
Download the Terraform Configs

Everything you need

Download Terraform configs using web

1. https://github.com/OktaEdu/es-dev-day-21
2. Download a ZIP
3. unzip





https://github.com/OktaEdu/es-delivery-dev

Use Git or checkout with SVN using the web URL.

Download Terraform configs using git

1. mkdir git
2. cd git
3. cd es-dev-day-21git clone --recursive https://github.com/OktaEdu/es-dev-day-21

```
1_org_spa_app
2_copy_clientid.sh
3_asa_project
4_copy_enrollment_token.sh
5_app_server
6_update_spa_app.sh
Scripts
LICENSE
README.md
```

Add Your Credentials to the Configs

1. Make a working copy of the git repo (avoid overwriting your changes)
2. Create an API token in your org with at least Org and App privileges
3. Update `1_org_spa_app/terraform.tfvars`
   - `org_name`
   - `base_url`
   - `api_token`
4. Create a service user in ASA with Team Admin role, add API key
5. Update scripts/set_asa_tf_vars.sh
   - `OKTAASA_KEY`
   - `OKTAASA_KEYSECRET`
   - `OKTAASA_TEAM`
   - Source it: . `Scripts/set_asa_tf_vars.sh`
6. Create AWS access keys, IAM user with only programmatic access, EC2 and VPC
7. Update 5_app_server/terraform.tfvars
   - `access_key`
   - `secret_key`
   - `org_url`
   - Note: `enrollment_token` and `clientid` are updated by scripts

- Run `chmod +x *.sh` if the scripts are not already executable.

04

Run the Configs

Configure your SPA app in your org

Create a new ASA project and provide access to the new app server

Provision a server using ASA AWS Starter Kit config

Configure your Okta Org

Commands	What it does
cd 1_org_spa_app/ terraform init terraform plan	Download okta Terraform provider Display plan of what will be created in your org. Review this plan
terraform apply	Updates your Org • Create the OAuth app • Create test users • Create a group with test users • Assign the group to the app
cd .. ./2_copy_clientid.sh	Copy the clientid for the new app to 5_app_server config/terraform.tfvars

Create a new Project in ASA

Commands	What it does
. scripts/set_asa_tf_vars.sh	Add ASA credentials vars to the environment Note: Source the file with "."
cd 3_asa_project terraform init terraform plan	Download the ASA Terraform provider Shows you the ASA team changes to be made. Review the plan
terraform apply	Updates your ASA Team • Creates ASA Project • Assigns group to project • Creates server enrollment token
cd .. ./4_copy_enrollment_token.sh	Copy the server enrollment token to 5_app_server config/terraform.tfvars

Deploy the Vue Sample App Server in AWS

Commands	What it does
cd 5_app_server terraform init terraform plan	Download the Terraform AWS provider Shows you the AWS changes to be made. Review the plan
terraform apply	Deploys your AWS Ubuntu instance • VPC subnet in two availability zones • Security group: ICMP, NTP, SSH, HTTPS • Ubuntu 16.04 latest • User data script: • ASA server agent • Bootstrap script for Vue Sample App
cd .. ./6_update_spa_app.sh	Update the org with the IP address of target server
cd 1_org_spa_app terraform apply	Update the Vue SPA app with new redirect_uris and post_logout_redirect_uris

05

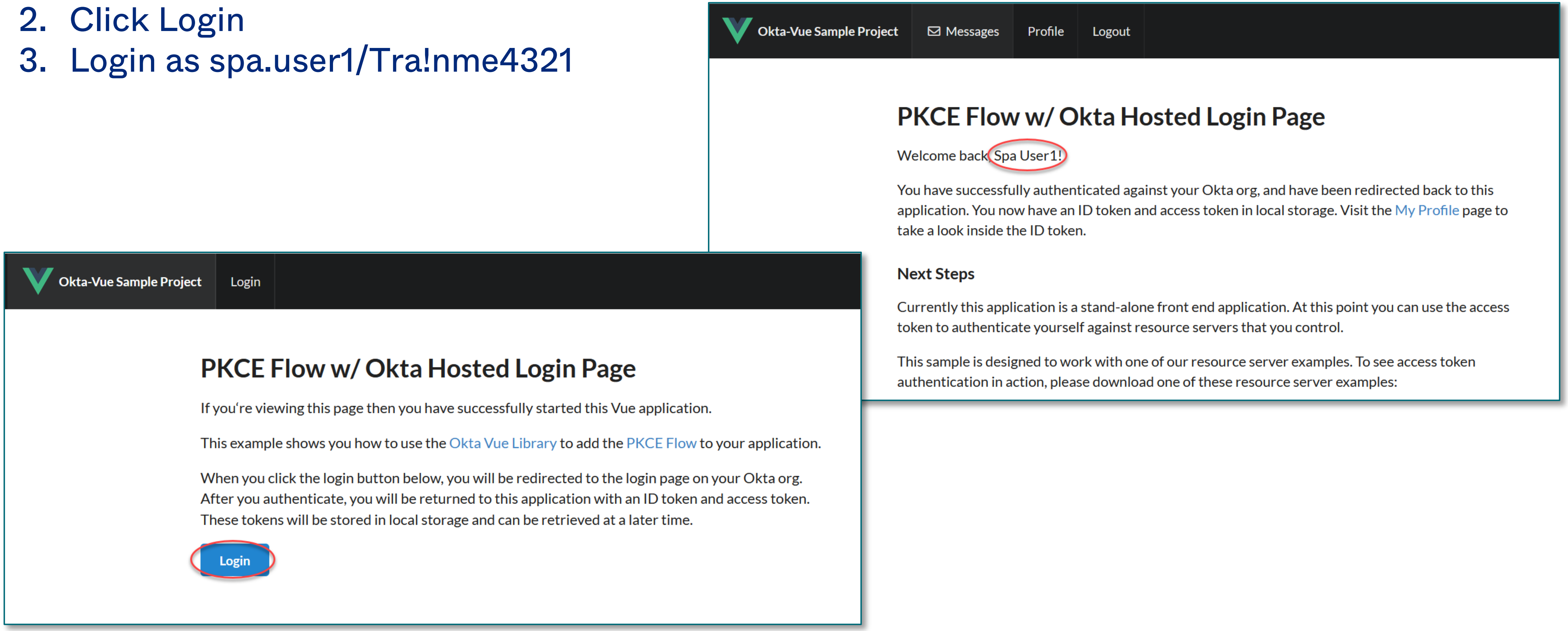
Test your App

Login to your App and Server

okta

Login to your Deployed App

1. https://yourserverip/
2. Click Login
3. Login as spa.user1/Tra!nme4321



PKCE Flow w/ Okta Hosted Login Page

If you're viewing this page then you have successfully started this Vue application.

This example shows you how to use the Okta Vue Library to add the PKCE Flow to your application.

When you click the login button below, you will be redirected to the login page on your Okta org. After you authenticate, you will be returned to this application with an ID token and access token. These tokens will be stored in local storage and can be retrieved at a later time.

PKCE Flow w/ Okta Hosted Login Page

Welcome back, Spa User1!

You have successfully authenticated against your Okta org, and have been redirected back to this application. You now have an ID token and access token in local storage. Visit the My Profile page to take a look inside the ID token.

Next Steps

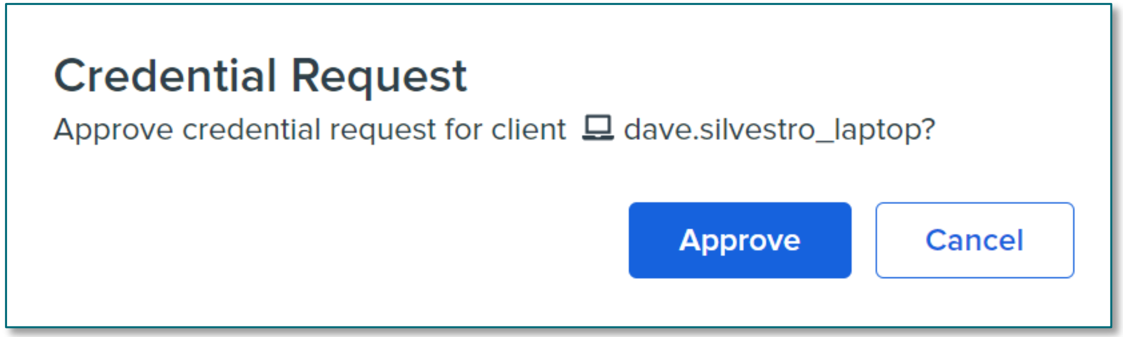
Currently this application is a stand-alone front end application. At this point you can use the access token to authenticate yourself against resource servers that you control.

This sample is designed to work with one of our resource server examples. To see access token authentication in action, please download one of these resource server examples:

okta

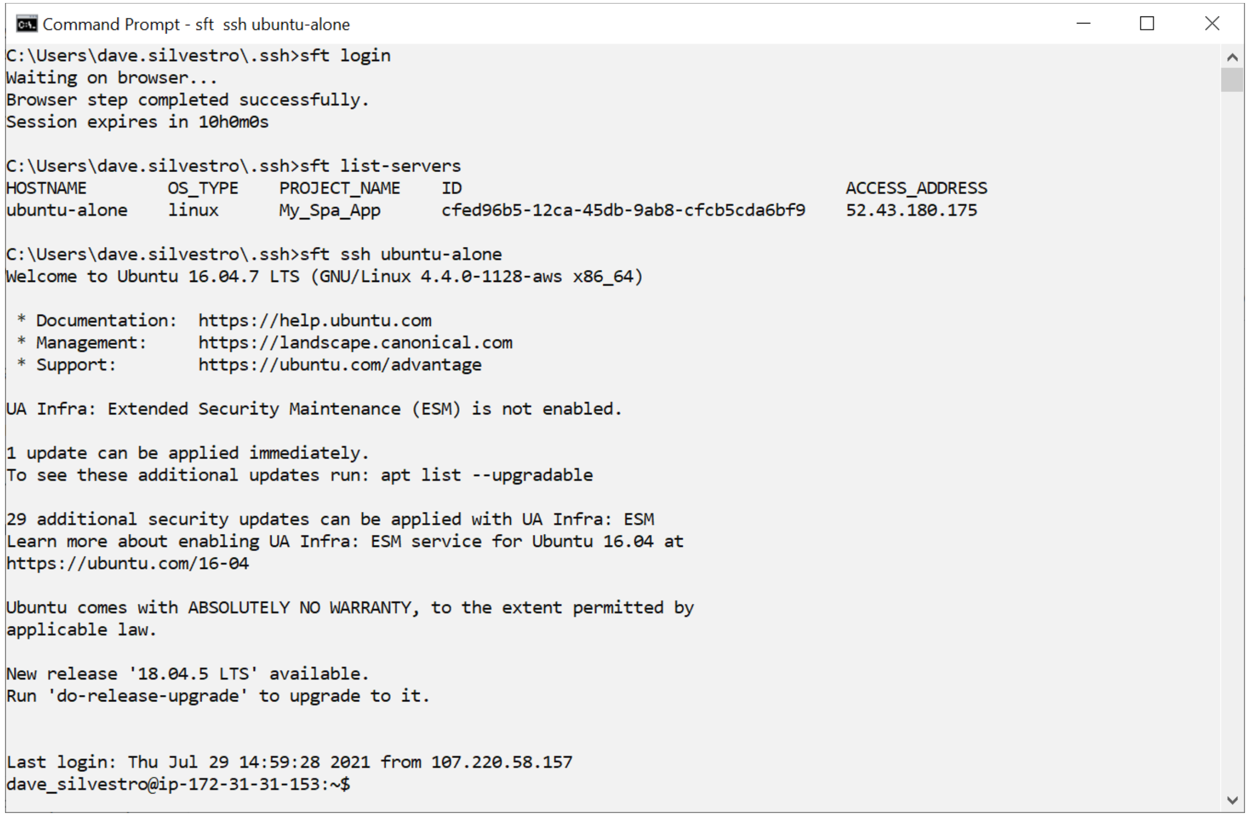
Check the Apache Logs on your App Server

1. If you have multiple ASA teams:
   - `sft list-accounts`
   - `sft use` *`account-id`*
2. `sft login`
3. Approve credential request
4. `sft list-servers`
5. `sft ssh ubuntu-alone`
   OR `sft putty ubuntu-alone`
6. `sudo tail -f \`
   `/var/log/apache2/error.log`



Credential Request

Approve credential request for client 💻 dave.silvestro_laptop?



```
C:\Users\dave.silvestro\.ssh>sft login
Waiting on browser...
Browser step completed successfully.
Session expires in 10h0m0s

C:\Users\dave.silvestro\.ssh>sft list-servers
HOSTNAME        OS_TYPE     PROJECT_NAME    ID                                      ACCESS_ADDRESS
ubuntu-alone    linux       My_Spa_App      cfed96b5-12ca-45db-9ab8-cfcb5cda6bf9    52.43.180.175

C:\Users\dave.silvestro\.ssh>sft ssh ubuntu-alone
Welcome to Ubuntu 16.04.7 LTS (GNU/Linux 4.4.0-1128-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

UA Infra: Extended Security Maintenance (ESM) is not enabled.

1 update can be applied immediately.
To see these additional updates run: apt list --upgradable

29 additional security updates can be applied with UA Infra: ESM
Learn more about enabling UA Infra: ESM service for Ubuntu 16.04 at
https://ubuntu.com/16-04

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

New release '18.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


Last login: Thu Jul 29 14:59:28 2021 from 107.220.58.157
dave_silvestro@ip-172-31-31-153:~$
```

okta

06

Destroy your Environment

Change your configs.

Rinse and Repeat!

Destroy the Environment

01

Changing only code base or target server config

1. cd 5_app_server
2. terraform destroy

02

Changing ASA settings, policy

1. cd 5_app_server
2. terraform destroy
3. cd ../3_asa_project
4. terraform destroy

03

Changing Okta settings, policy

1. cd 5_app_server
2. terraform destroy
3. cd ../3_asa_project
4. terraform destroy
5. cd ../1_org_spa_app
6. terraform destroy

Change your Configs

01

Use your own bootstrap script

1. Create a new GitHub repo
2. Change the user-data script

02

Use a modified version of the Vue Sample Code

1. Create a new GitHub repo
2. Change AWS terraform.tfvars

03

Add a resource server!

- Node/Express Resource Server Example
- Java/Spring MVC Resource Server Example
- ASP.NET Core Web API Resource Server Example

Key takeaways

01

Okta Sample Apps give you a head start for your application authentication needs

02

Plan for your target environment:

- App settings
- Target server
- App and server access policy

03

Leverage the Terraform providers for orgs and ASA to automate your deployment

Learn Okta and Certify your Skills

Training Courses

Okta Customer Identity for Developers

Implement Advanced Server Access

Resources for this Lab

Vue Sample Application for Okta

Lab Terraform Configs

ASA StarterKit for AWS, Ubuntu

```
./scripts/get_asa_starter_kit.sh
```

Certification

Certified Developer Study Guide

Register for courses at okta.com/training and visit
developer.okta.com for guides, docs, and sample code