» auth0_client_grant

Auth0 uses various grant types, or methods by which you grant limited access to your resources to another entity without exposing credentials. The OAuth 2.0 protocol supports several types of grants, which allow different types of access. This resource allows you to create and manage client grants used with configured Auth0 clients.

» Example Usage

```
resource "auth0_client" "my_client" {
  name = "Example Application - Client Grant (Managed by Terraform)"
}

resource "auth0_resource_server" "my_resource_server" {
  name       = "Example Resource Server - Client Grant (Managed by Terraform)"
  identifier = "https://api.example.com/client-grant"

  scopes {
    value       = "create:foo"
    description = "Create foos"
  }

  scopes {
    value       = "create:bar"
    description = "Create bars"
  }
}

resource "auth0_client_grant" "my_client_grant" {
  client_id = "${auth0_client.my_client.id}"
  audience  = "${auth0_resource_server.my_resource_server.identifier}"
  scope     = ["create:foo"]
}
```

» Argument Reference

Arguments accepted by this resource include:

- `client_id` - (Required) String. ID of the client for this grant.
- `audience` - (Required) String. Audience or API Identifier for this grant.
- `scope` - (Required) List(String). Permissions (scopes) included in this grant.

» auth0_client

With this resource, you can set up applications that use Auth0 for authentication and configure allowed callback URLs and secrets for these applications. Depending on your plan, you may also configure add-ons to allow your application to call another application's API (such as Firebase and AWS) on behalf of an authenticated user.

» Example Usage

```
resource "auth0_client" "my_client" {
  name = "Application - Acceptance Test"
  description = "Test Applications Long Description"
  app_type = "non_interactive"
  custom_login_page_on = true
  is_first_party = true
  is_token_endpoint_ip_header_trusted = true
  token_endpoint_auth_method = "client_secret_post"
  oidc_conformant = false
  callbacks = [ "https://example.com/callback" ]
  allowed_origins = [ "https://example.com" ]
  grant_types = [ "authorization_code", "http://auth0.com/oauth/grant-type/password-realm",
  allowed_logout_urls = [ "https://example.com" ]
  web_origins = [ "https://example.com" ]
  jwt_configuration {
    lifetime_in_seconds = 300
    secret_encoded = true
    alg = "RS256"
    scopes = {
      foo = "bar"
    }
  }
  client_metadata = {
    foo = "zoo"
  }
  addons {
    firebase = {
      client_email = "john.doe@example.com"
      lifetime_in_seconds = 1
      private_key = "wer"
      private_key_id = "qwreerwerwe"
    }
    samlp {
      audience = "https://example.com/saml"
```

```
      mappings = {
        email = "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress"
        name = "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name"
      }
      create_upn_claim = false
      passthrough_claims_with_no_mapping = false
      map_unknown_claims_as_is = false
      map_identities = false
      name_identifier_format = "urn:oasis:names:tc:SAML:2.0:nameid-format:persistent"
      name_identifier_probes = [
        "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress"
      ]
    }
  }
  mobile {
    ios {
      team_id = "9JA89QQLNQ"
      app_bundle_identifier = "com.my.bundle.id"
    }
  }
}
```

» Argument Reference

Arguments accepted by this resource include:

- `name` - (Required) String. Name of the client.
- `description` - (Optional) String, (Max length = 140 characters). Description of the purpose of the client.
- `client_secret_rotation_trigger` - (Optional) Map.
- `app_type` - (Optional) String. Type of application the client represents. Options include `native`, `spa`, `regular_web`, `non_interactive`, `rms`, `box`, `cloudbees`, `concur`, `dropbox`, `mscrm`, `echosign`, `egnyte`, `newrelic`, `office365`, `salesforce`, `sentry`, `sharepoint`, `slack`, `springcm`, `zendesk`, `zoom`.
- `logo_uri` - (Optional) String. URL of the logo for the client. Recommended size is 150px x 150px. If none is set, the default badge for the application type will be shown.
- `is_first_party` - (Optional) Boolean. Indicates whether or not this client is a first-party client.
- `is_token_endpoint_ip_header_trusted` - (Optional) Boolean. Indicates whether or not the token endpoint IP header is trusted.
- `oidc_conformant` - (Optional) Boolean. Indicates whether or not this client will conform to strict OIDC specifications.
- `callbacks` - (Optional) List(String). URLs that Auth0 may call back

to after a user authenticates for the client. Make sure to specify the protocol (https://) otherwise the callback may fail in some cases. With the exception of custom URI schemes for native clients, all callbacks should use protocol https://.

- `allowed_logout_urls` - (Optional) List(String). URLs that Auth0 may redirect to after logout.
- `grant_types` - (Optional) List(String). Types of grants that this client is authorized to use.
- `allowed_origins` - (Optional) List(String). URLs that represent valid origins for cross-origin resource sharing. By default, all your callback URLs will be allowed.
- `web_origins` - (Optional) List(String). URLs that represent valid web origins for use with web message response mode.
- `jwt_configuration` - (Optional) List(Resource). Configuration settings for the JWTs issued for this client. For details, see JWT Configuration.
- `encryption_key` - (Optional) Map(String).
- `sso` - (Optional) Boolean. Indicates whether or not the client should use Auth0 rather than the IdP to perform Single Sign-On (SSO). True = Use Auth0.
- `sso_disabled` - (Optional) Boolean. Indicates whether or not SSO is disabled.
- `cross_origin_auth` - (Optional) Boolean. Indicates whether or not the client can be used to make cross-origin authentication requests.
- `cross_origin_loc` - (Optional) String. URL for the location on your site where the cross-origin verification takes place for the cross-origin auth flow. Used when performing auth in your own domain instead of through the Auth0-hosted login page.
- `custom_login_page_on` - (Optional) Boolean. Indicates whether or not a custom login page is to be used.
- `custom_login_page` - (Optional) String. Content of the custom login page.
- `custom_login_page_preview` - (Optional) String.
- `form_template` - (Optional) String. Form template for WS-Federation protocol.
- `addons` - (Optional) List(Resource). Configuration settings for add-ons for this client. For details, see Add-ons.
- `token_endpoint_auth_method` - (Optional) String. Defines the requested authentication method for the token endpoint. Options include `none` (public client without a client secret), `client_secret_post` (client uses HTTP POST parameters), `client_secret_basic` (client uses HTTP Basic).
- `client_metadata` - (Optional) Map(String)
- `mobile` - (Optional) List(Resource). Configuration settings for mobile native applications. For details, see Mobile.

» JWT Configuration

`jwt_configuration` supports the following arguments:

- `lifetime_in_seconds` - (Optional) Integer. Number of seconds during which the JWT will be valid.
- `secret_encoded` - (Optional) Boolean. Indicates whether or not the client secret is base64 encoded.
- `scopes` - (Optional) Map(String). Permissions (scopes) included in JWTs.
- `alg` - (Optional) String. Algorithm used to sign JWTs.

» Add-ons

`addons` supports the following arguments:

- `aws` - (Optional) String
- `azure_blob` - (Optional) String
- `azure_sb` - (Optional) String
- `box` - (Optional) String
- `cloudbees` - (Optional) String
- `concur` - (Optional) String
- `dropbox`- (Optional) String
- `echosign`- (Optional) String
- `egnyte`- (Optional) String
- `firebase`- (Optional) String
- `mscrm` - (Optional) String
- `newrelic`- (Optional) String
- `office365`- (Optional) String
- `rms` - (Optional) String
- `salesforce`- (Optional) String
- `salesforce_api`- (Optional) String
- `salesforce_sandbox_api`- (Optional) String
- `samlp` - (Optional) List(Resource). Configuration settings for a SAML add-on. For details, see SAML.
- `layer`- (Optional) String
- `sap_api`- (Optional) String
- `sentry` - (Optional) String
- `sharepoint`- (Optional) String
- `slack` - (Optional) String
- `springcm`- (Optional) String
- `wams`- (Optional) String
- `wsfed`- (Optional) String
- `zendesk`- (Optional) String
- `zoom`- (Optional) String

» SAML

`samlp` supports the following arguments:

- `audience` - (Optional) String. Audience of the SAML Assertion. Default will be the Issuer on SAMLRequest.
- `recipient` - (Optional) String. Recipient of the SAML Assertion (SubjectConfirmationData). Default is AssertionConsumerUrl on SAMLRequest or Callback URL if no SAMLRequest was sent.
- `create_upn_claim` - (Optional) Boolean, (Default=true) Indicates whether or not a UPN claim should be created.
- `passthrough_claims_with_no_mapping` - (Optional) Boolean, (Default=true). Indicates whether or not to passthrough claims that are not mapped to the common profile in the output assertion.
- `map_unknown_claims_as_is` - (Optional) Boolean, (Default=false). Indicates whether or not to add a prefix of `http://schema.auth0.com` to any claims that are not mapped to the common profile when passed through in the output assertion.
- `map_identities` - (Optional) Boolean, (Default=true). Indicates whether or not to add additional identity information in the token, such as the provider used and the access_token, if available.
- `signature_algorithm` - (Optional) String, (Default=`rsa-sha1`). Algorithm used to sign the SAML Assertion or response. Options include `rsa-sha1` and `rsa-sha256`.
- `digest_algorithm` - (Optional) String, (Default=`sha1`). Algorithm used to calculate the digest of the SAML Assertion or response. Options include `defaultsha1` and `sha256`.
- `destination` - (Optional) String. Destination of the SAML Response. If not specified, it will be AssertionConsumerUrlof SAMLRequest or Callback URL if there was no SAMLRequest.
- `lifetime_in_seconds` - (Optional) Integer, (Default=3600). Number of seconds during which the token is valid.
- `sign_response` - (Optional) Boolean. Indicates whether or not the SAML Response should be signed instead of the SAML Assertion.
- `typed_attributes` - (Optional) Boolean, (Default=true). Indicates whether or not we should infer the `xs:type` of the element. Types include `xs:string`, `xs:boolean`, `xs:double`, and `xs:anyType`. When set to false, all `xs:type` are `xs:anyType`.
- `include_attribute_name_format` - (Optional) Boolean,(Default=true). Indicates whether or not we should infer the NameFormat based on the attribute name. If set to false, the attribute NameFormat is not set in the assertion.
- `name_identifier_format` - (Optional) String, (Default=`urn:oasis:names:tc:SAML:1.1:nameid-forma` Format of the name identifier.
- `authn_context_class_ref` - (Optional) String. Class reference of the authentication context.

- `binding` - (Optional) String. Protocol binding used for SAML logout responses.
- `mappings` - (Optional) Map(String). Mappings between the Auth0 user profile property name (`name`) and the output attributes on the SAML attribute in the assertion (`value`).
- `logout` - (Optional) Map(Resource). Configuration settings for logout. For details, see Logout.
- `name_identifier_probes` - (Optional) List(String). Attributes that can be used for Subject/NameID. Auth0 will try each of the attributes of this array in order and use the first value it finds.

» Logout

`logout` supports the following options:

- `callback` - (Optional) String. Service provider's Single Logout Service URL, to which Auth0 will send logout requests and responses.
- `slo_enabled` - (Optional) Boolean. Indicates whether or not Auth0 should notify service providers of session termination.

» Mobile

`mobile` supports the following arguments:

- `android` (Optional) List(Resource). Configuration settings for Android native apps. For details, see Android.
- `ios` (Optional) List(Resource). Configuration settings for i0S native apps. For details, see iOS.

» Android

`android` supports the following arguments:

- `app_package_name` (Optional) String
- `sha256_cert_fingerprints` (Optional) List(String)

» iOS

`ios` supports the following arguments:

- `team_id` - (Optional) String
- `app_bundle_identifier` - (Optional) String

» Attribute Reference

Attributes exported by this resource include:

- `client_id` - String. ID of the client.
- `client_secret` - String. Secret for the client; keep this private.
- `is_first_party` - Boolean. Indicates whether or not this client is a first-party client.
- `is_token_endpoint_ip_header_trusted` - Boolean
- `oidc_conformant` - Boolean. Indicates whether or not this client will conform to strict OIDC specifications.
- `grant_types` - List(String). Types of grants that this client is authorized to use.
- `custom_login_page_on` - Boolean. Indicates whether or not a custom login page is to be used.
- `token_endpoint_auth_method` - String. Defines the requested authentication method for the token endpoint. Options include `none` (public client without a client secret), `client_secret_post` (client uses HTTP POST parameters), `client_secret_basic` (client uses HTTP Basic).

» auth0_connection

With Auth0, you can define sources of users, otherwise known as connections, which may include identity providers (such as Google or LinkedIn), databases, or passwordless authentication methods. This resource allows you to configure and manage connections to be used with your clients and users.

» Example Usage

```
resource "auth0_connection" "my_connection" {
  name = "Example-Connection"
  strategy = "auth0"
  options {
    password_policy = "excellent"
    password_history {
      enable = true
      size = 3
    }
    brute_force_protection = "true"
    enabled_database_customization = "true"
    custom_scripts = {
      get_user = <<EOF
function getByEmail (email, callback) {
  return callback(new Error("Whoops!"))
```

```
}
EOF
    }

    configuration = {
      foo = "bar"
      bar = "baz"
    }
  }
}

resource "auth0_connection" "my_waad_connection" {
   name     = "my-waad-connection"
   strategy = "waad"

   options {
     client_id     = "1234"
     client_secret = "1234"
     tenant_domain = "exmaple.onmicrosoft.com"

     domain_aliases = [
       "example.io",
     ]

     use_wsfed            = false
     waad_protocol        = "openid-connect"
     waad_common_endpoint = false

     app_domain       = "my-auth0-app.eu.auth0.com"
     api_enable_users = true
     basic_profile    = true
     ext_groups       = true
     ext_profile      = true
   }
 }
```

» Argument Reference

Arguments accepted by this resource include:

- `name` - (Required) String. Name of the connection.
- `is_domain_connection` - (Optional) Boolean. Indicates whether or not the connection is domain level.
- `strategy` - (Required) String. Type of the connection, which indicates the identity provider. Options include `ad`, `adfs`, `amazon`,

`aol`, `apple`, `auth0`, `auth0-adldap`, `auth0-oidc`, `baidu`, `bitbucket`, `bitly`, `box`, `custom`, `daccount`, `dropbox`, `dwolla`, `email`, `evernote`, `evernote-sandbox`, `exact`, `facebook`, `fitbit`, `flickr`, `github`, `google-apps`, `google-oauth2`, `guardian`, `instagram`, `ip`, `line`, `linkedin`, `miicard`, `oauth1`, `oauth2`, `office365`, `oidc`, `paypal`, `paypal-sandbox`, `pingfederate`, `planningcenter`, `renren`, `salesforce`, `salesforce-community`, `salesforce-sandbox` `samlp`, `sharepoint`, `shopify`, `sms`, `soundcloud`, `thecity`, `thecity-sandbox`, `thirtysevensignals`, `twitter`, `untappd`, `vkontakte`, `waad`, `weibo`, `windowslive`, `wordpress`, `yahoo`, `yammer`, `yandex`.

- `options` - (Optional) List(Resource). Configuration settings for connection options. For details, see Options.
- `enabled_clients` - (Optional) Set(String). IDs of the clients for which the connection is enabled. If not specified, no clients are enabled.
- `realms` - (Optional) List(String). Defines the realms for which the connection will be used (i.e., email domains). If not specified, the connection name is added as the realm.

» Options

`options` supports the following arguments:

- `validation` - (Optional) String.
- `password_policy` - (Optional) String. Indicates level of password strength to enforce during authentication. A strong password policy will make it difficult, if not improbable, for someone to guess a password through either manual or automated means. Options include `none`, `low`, `fair`, `good`, `excellent`.
- `password_history` - (Optional) List(Resource). Configuration settings for the password history that is maintained for each user to prevent the reuse of passwords. For details, see Password History.
- `password_no_personal_info` - (Optional) List(Resource). Configuration settings for the password personal info check, which does not allow passwords that contain any part of the user's personal data, including user's name, username, nickname, user_metadata.name, user_metadata.first, user_metadata.last, user's email, or firstpart of the user's email. For details, see Password No Personal Info.
- `password_dictionary` - (Optional) List(Resource). Configuration settings for the password dictionary check, which does not allow passwords that are part of the password dictionary. For details, see Password Dictionary.
- `password_complexity_options` - (Optional) List(Resource). Configuration settings for password complexity. For details, see Password Complexity Options.
- `api_enable_users` - (Optional) Boolean.

- `basic_profile` - (Optional) Boolean.
- `ext_admin` - (Optional) Boolean.
- `ext_is_suspended` - (Optional) Boolean.
- `ext_agreed_terms` - (Optional) Boolean.
- `ext_groups` - (Optional) Boolean.
- `ext_nested_groups` - (Optional) Boolean.
- `ext_assigned_plans` - (Optional) Boolean.
- `ext_profile` - (Optional) Boolean.
- `enabled_database_customization` - (Optional) Boolean.
- `brute_force_protection` - (Optional) Boolean. Indicates whether or not to enable brute force protection, which will limit the number of signups and failed logins from a suspicious IP address.
- `import_mode` - (Optional) Boolean. Indicates whether or not you have a legacy user store and want to gradually migrate those users to the Auth0 user store. Learn more.
- `disable_signup` - (Optional) Boolean. Indicates whether or not to allow user sign-ups to your application.
- `requires_username` - (Optional) Boolean. Indicates whether or not the user is required to provide a username in addition to an email address.
- `custom_scripts` - (Optional) Map(String).
- `configuration` - (Optional) Map(String), Case-sensitive.

Azure AD Options

- `app_id` - (Optional) String
- `app_domain` - (Optional) String. Azure AD domain name.
- `client_id` - (Optional) String. Client ID for your Azure AD application.
- `client_secret` - (Optional) String, Case-sensitive. Client secret for your Azure AD application.
- `domain_aliases` - (Optional) List(String). List of the domains that can be authenticated using the Identity Provider. Only needed for Identifier First authentication flows.
- `max_groups_to_retrieve` - (Optional) String. Maximum number of groups to retrieve.
- `tenant_domain` - (Optional) String
- `use_wsfed` - (Optional) Bool
- `waad_protocol` - (Optional) String
- `waad_common_endpoint` - (Optional) Boolean. Indicates whether or not to use the common endpoint rather than the default endpoint. Typically enabled if you're using this for a multi-tenant application in Azure AD.

Twilio/SMS Options

- `name` - (Optional) String.
- `twilio_sid` - (Optional) String. SID for your Twilio account.
- `twilio_token` - (Optional) String, Case-sensitive. AuthToken for your Twilio account.

- `from` - (Optional) String. SMS number for the sender. Used when SMS Source is From.
- `syntax` - (Optional) String. Syntax of the SMS. Options include `markdown` and `liquid`.
- `template` - (Optional) String. Template for the SMS. You can use `@@password@@` as a placeholder for the password value.
- `totp` - (Optional) Map(Resource). Configuration options for one-time passwords. For details, see TOTP.
- `messaging_service_sid` - (Optional) String. SID for Copilot. Used when SMS Source is Copilot.

ADFS Options

- `adfs_server` - (Optional) String. ADFS Metadata source.

Salesforce Options

- `community_base_url` - (Optional) String.

» Password History

`password_history` supports the following arguments:

- `enable` - (Optional) Boolean. Indicates whether password history is enabled for the connection. When enabled, any existing users in this connection will be unaffected; the system will maintain their password history going forward.
- `size` - (Optional) Integer, (Maximum=24). Indicates the number of passwords to keep in history.

» Password No Personal Info

`password_no_personal_info` supports the following arguments:

- `enable` - (Optional) Boolean. Indicates whether the password personal info check is enabled for this connection.

» Password Dictionary

`passsword_dictionary` supports the following arguments:

- `enable` - (Optional) Boolean. Indicates whether the password dictionary check is enabled for this connection.
- `dictionary` - (Optional) Set(String), (Maximum=2000 characters). Customized contents of the password dictionary. By default, the password dictionary contains a list of the 10,000 most common passwords; your customized content is used in addition to the default password dictionary. Matching is not case-sensitive.

» Password Complexity Options

`password_complexity_options` supports the following arguments:

- `min_length`- (Optional) Integer. Minimum number of characters allowed in passwords.

» TOTP

`totp` supports the following arguments:

- `time_step` - (Optional) Integer. Seconds between allowed generation of new passwords.
- `length` - (Optional) Integer. Length of the one-time password.

» Attribute Reference

Attributes exported by this resource include:

- `is_domain_connection` - Boolean. Indicates whether or not the connection is domain level.
- `options` - List(Resource). Configuration settings for connection options. For details, see Options Attributes.
- `realms` - List(String). Defines the realms for which the connection will be used (i.e., email domains). If the array is empty or the property is not specified, the connection name is added as the realm.

» Options Attributes

`options` exports the following attributes:

- `password_history` - List(Resource). Configuration settings for the password history that is maintained for each user to prevent the reuse of passwords. For details, see Password History Attributes.

» Password History Attributes

`password_history` exports the following attributes:

- `enable` - Boolean. Indicates whether password history is enabled for the connection. When enabled, any existing users in this connection will be unaffected; the system will maintain their password history going forward.
- `size` - Integer. Indicates the number of passwords to keep in history.

» auth0_custom_domain

With Auth0, you can use a custom domain to maintain a consistent user experience. This resource allows you to create and manage a custom domain within your Auth0 tenant.

» Example Usage

```
resource "auth0_custom_domain" "my_custom_domain" {
  domain = "auth.example.com"
  type = "auth0_managed_certs"
  verification_method = "txt"
}
```

» Argument Reference

Arguments accepted by this resource include:

- `domain` - (Required) String. Name of the custom domain.
- `type` - (Required) String. Provisioning type for the custom domain. Options include `auth0_managed_certs` and `self_managed_certs`.
- `verification_method` - (Required) String. Domain verification method. Options include `txt`.

» Attribute Reference

Attributes exported by this resource include:

- `primary` - Boolean. Indicates whether or not this is a primary domain.
- `status` - String. Configuration status for the custom domain. Options include `disabled`, `pending`, `pending_verification`, and `ready`.
- `verification` - List(Resource). Configuration settings for verification. For details, see Verification.

» Verification

`verification` exports the following attributes:

- `methods` - List(Map). Verification methods for the domain.

» auth0_email_template

With Auth0, you can have standard welcome, password reset, and account verification email-based workflows built right into Auth0. This resource allows you to configure email templates to customize the look, feel, and sender identities of emails sent by Auth0. Used in conjunction with configured email providers.

» Example Usage

```
resource "auth0_email" "my_email_provider" {
  name = "ses"
  enabled = true
  default_from_address = "accounts@example.com"
  credentials {
    access_key_id = "AKIAXXXXXXXXXXXXXXXX"
    secret_access_key = "7e8c2148xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
    region = "us-east-1"
  }
}

resource "auth0_email_template" "my_email_template" {
  template = "welcome_email"
  body = "<html><body><h1>Welcome!</h1></body></html>"
  from = "welcome@example.com"
  result_url = "https://example.com/welcome"
  subject = "Welcome"
  syntax = "liquid"
  url_lifetime_in_seconds = 3600
  enabled = true

  depends_on = [ "${auth0_email.my_email_provider}" ]
}
```

» Argument Reference

Arguments accepted by this resource include:

- `template` - (Required) String. Template name. Options include `verify_email`, `reset_email`, `welcome_email`, `blocked_account`, `stolen_credentials`, `enrollment_email`, `mfa_oob_code`, `change_password` (legacy), and `password_reset` (legacy).
- `body` - (Required) String. Body of the email template. You can include common variables.

- `from` - (Required) String. Email address to use as the sender. You can include common variables.
- `result_url` - (Required) String. URL to redirect the user to after a successful action. Learn more.
- `subject` - (Required) String. Subject line of the email. You can include common variables.
- `syntax` - (Required) String. Syntax of the template body. You can use either text or HTML + Liquid syntax.
- `url_lifetime_in_seconds` - (Optional) Integer. Number of seconds during which the link within the email will be valid.
- `enabled` - (Required) Boolean. Indicates whether or not the template is enabled.

» auth0_email

With Auth0, you can have standard welcome, password reset, and account verification email-based workflows built right into Auth0. This resource allows you to configure email providers so you can route all emails that are part of Auth0's authentication workflows through the supported high-volume email service of your choice.

» Example Usage

```
resource "auth0_email" "my_email_provider" {
  name = "ses"
  enabled = true
  default_from_address = "accounts@example.com"
  credentials {
    access_key_id = "AKIAXXXXXXXXXXXXXXX"
    secret_access_key = "7e8c2148xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
    region = "us-east-1"
  }
}
```

» Argument Reference

Arguments accepted by this resource include:

- `name` - (Required) String. Name of the email provider. Options include `mailgun`, `mandrill`, `sendgrid`, `ses`, `smtp`, and `sparkpost`.
- `enabled` - (Optional) Boolean. Indicates whether or not the email provider is enabled.

- `default_from_address` - (Required) String. Email address to use as the sender when no other "from" address is specified.
- `credentials` - (Required) List(Resource). Configuration settings for the credentials for the email provider. For details, see Credentials.

» Credentials

`credentials` supports the following arguments:

- `api_user` - (Optional) String. API User for your email service.
- `api_key` - (Optional) String, Case-sensitive. API Key for your email service. Will always be encrypted in our database.
- `access_key_id` - (Optional) String, Case-sensitive. AWS Access Key ID. Used only for AWS.
- `secret_access_key` - (Optional) String, Case-sensitive. AWS Secret Key. Will always be encrypted in our database. Used only for AWS.
- `region` - (Optional) String. Default region. Used only for AWS, Mailgun, and SparkPost.
- `smtp_host` - (Optional) String. Hostname or IP address of your SMTP server. Used only for SMTP.
- `smtp_port` - (Optional) Integer. Port used by your SMTP server. Please avoid using port 25 if possible because many providers have limitations on this port. Used only for SMTP.
- `smtp_user` - (Optional) String. SMTP username. Used only for SMTP.
- `smtp_pass` - (Optional) String, Case-sensitive. SMTP password. Used only for SMTP.

» auth0_resource_server

With this resource, you can set up APIs that can be consumed from your authorized applications.

» Example Usage

```
resource "auth0_resource_server" "my_resource_server" {
  name        = "Example Resource Server (Managed by Terraform)"
  identifier  = "https://api.example.com"
  signing_alg = "RS256"

  scopes {
    value       = "create:foo"
    description = "Create foos"
  }
```

```
  scopes {
    value       = "create:bar"
    description = "Create bars"
  }

  allow_offline_access                              = true
  token_lifetime                                    = 8600
  skip_consent_for_verifiable_first_party_clients = true
}
```

» Argument Reference

Arguments accepted by this resource include:

- `name` - (Optional) String. Friendly name for the resource server. Cannot include `<` or `>` characters.
- `identifier` - (Optional) String. Unique identifier for the resource server. Used as the audience parameter for authorization calls. Can not be changed once set.
- `scopes` - (Optional) Set(Resource). List of permissions (scopes) used by this resource server. For details, see Scopes.
- `signing_alg` - (Optional) String. Algorithm used to sign JWTs. Options include `HS256` and `RS256`.
- `signing_secret` - (Optional) String. Secret used to sign tokens when using symmetric algorithms (HS256).
- `allow_offline_access` - (Optional) Boolean. Indicates whether or not refresh tokens can be issued for this resource server.
- `token_lifetime` - (Optional) Integer. Number of seconds during which access tokens issued for this resource server from the token endpoint remain valid.
- `token_lifetime_for_web` - (Optional) Integer. Number of seconds during which access tokens issued for this resource server via implicit or hybrid flows remain valid. Cannot be greater than the `token_lifetime` value.
- `skip_consent_for_verifiable_first_party_clients` - (Optional) Boolean. Indicates whether or not to skip user consent for applications flagged as first party.
- `verification_location` - (Optional) String
- `options` - (Optional) Map(String). Used to store additional metadata
- `enforce_policies` - (Optional) Boolean. Indicates whether or not authorization polices are enforced.
- `token_dialect` - (Optional) String. Dialect of access tokens that should be issued for this resource server. Options include `access_token` or `access_token_authz` (includes permissions).

» Scopes

`scopes` supports the following arguments:

- `value` - (Optional) String. Name of the permission (scope). Examples include `read:appointments` or `delete:appointments`.
- `description` - (Optional) String. Description of the permission (scope).

» Attribute Reference

Attributes exported by this resource include:

- `signing_alg` - String. Algorithm used to sign JWTs. Options include `HS256` and `RS256`.
- `signing_secret` - String. Secret used to sign tokens when using symmetric algorithms (HS256).
- `token_lifetime` - Integer. Number of seconds during which access tokens issued for this resource server from the token endpoint remain valid.
- `token_lifetime_for_web` - Integer. Number of seconds during which access tokens issued for this resource server via implicit or hybrid flows remain valid. Cannot be greater than the `token_lifetime` value.

» auth0_role

With this resource, you can created and manage collections of permissions that can be assigned to users, which are otherwise known as roles. Permissions (scopes) are created on auth0_resource_server, then associated with roles and optionally, users using this resource.

» Example Usage

```
resource "auth0_resource_server" "my_resource_server" {
  name = "My Resource Server (Managed by Terraform)"
  identifier = "my-resource-server-identifier"
  signing_alg = "RS256"
  token_lifetime = 86400
  skip_consent_for_verifiable_first_party_clients = true

  enforce_policies = true

  scopes {
    value = "read:something"
    description = "read something"
```

```
  }
}

resource "auth0_user" "my_user" {
  connection_name = "Username-Password-Authentication"
  user_id = "auth0|1234567890"
  email = "test@test.com"
  password = "passpass$12$12"
  nickname = "testnick"
  username = "testnick"
  roles = [ "${auth0_role.my_role.id}" ]
}
resource "auth0_role" "my_role" {
  name = "My Role - (Managed by Terraform)"
  description = "Role Description..."

  permissions {
    resource_server_identifier = "${auth0_resource_server.my_resource_server.identifier}"
    name = "read:something"
  }
}
```

» Argument Reference

Arguments accepted by this resource include:

- `role_id` - (Optional) String. ID for this role.
- `name` - (Required) String. Name for this role.
- `description` - (Optional) String. Description of the role.
- `user_ids` - (Optional) List(String). IDs of the users to which the role is assigned.
- `permissions` - (Optional) Set(Resource). Configuration settings for permissions (scopes) attached to the role. For details, see Permissions.

» Permissions

`permissions` supports the following arguments:

- `name` - (Required) String. Name of the permission (scope).
- `resource_server_identifier` - (Required) String. Unique identifier for the resource server.

» Attribute Reference

Attributes exported by this resource include:

- `role_id` - String. ID for the role.

» auth0_rule_config

With Auth0, you can create custom Javascript snippets that run in a secure, isolated sandbox as part of your authentication pipeline, which are otherwise known as rules. This resource allows you to create and manage variables that are available to all rules via Auth0's global configuration object. Used in conjunction with configured rules.

» Example Usage

```
resource "auth0_rule" "my_rule" {
  name = "empty-rule"
  script = <<EOF
function (user, context, callback) {
  callback(null, user, context);
}
EOF
  enabled = true
}

resource "auth0_rule_config" "my_rule_config" {
  key = "foo"
  value = "bar"
}
```

» Argument Reference

Arguments accepted by this resource include:

- `key` - (Required) String. Key for a rules configuration variable.
- `value` - (Required) String, Case-sensitive. Value for a rules configuration variable.

» auth0_rule

With Auth0, you can create custom Javascript snippets that run in a secure, isolated sandbox as part of your authentication pipeline, which are otherwise known as rules. This resource allows you to create and manage rules. You can create global variable for use with rules by using the auth0_rule_config resource.

» Example Usage

```
resource "auth0_rule" "my_rule" {
  name = "empty-rule"
  script = <<EOF
function (user, context, callback) {
  callback(null, user, context);
}
EOF
  enabled = true
}

resource "auth0_rule_config" "my_rule_config" {
  key = "foo"
  value = "bar"
}
```

» Argument Reference

Arguments accepted by this resource include:

- `name` - (Required) String. Name of the rule. May only contain alphanumeric characters, spaces, and hyphens. May neither start nor end with hyphens or spaces.
- `script` - (Required) String. Code to be executed when the rule runs.
- `order` - (Optional) Integer. Order in which the rule executes relative to other rules. Lower-valued rules execute first.
- `enabled` - (Optional) Boolean. Indicates whether the rule is enabled.

» Attribute Reference

Attributes exported by this resource include:

- `order` - Integer. Order in which the rule executes relative to other rules. Lower-valued rules execute first.

» auth0_hook

Hooks are secure, self-contained functions that allow you to customize the behavior of Auth0 when executed for selected extensibility points of the Auth0 platform. Auth0 invokes Hooks during runtime to execute your custom Node.js code.

Depending on the extensibility point, you can use Hooks with Database Connections and/or Passwordless Connections.

» Example Usage

```
resource "auth0_hook" "my_hook" {
  name = "My Pre User Registration Hook"
  script = <<EOF
function (user, context, callback) {
  callback(null, { user });
}
EOF
  trigger_id = "pre-user-registration"
  enabled = true
}
```

» Argument Reference

The following arguments are supported:

- `enabled` - (Optional) Whether the hook is enabled, or disabled
- `name` - (Required) Name of this hook
- `script` - (Required) Code to be executed when this hook runs
- `trigger_id` - (Required) Execution stage of this rule. Can be credentials-exchange, pre-user-registration, post-user-registration, post-change-password, or send-phone-message

» auth0_prompt

With this resource, you can manage your Auth0 prompts, including choosing the login experience version.

» Example Usage

```
resource "auth0_prompt" "example" {
```

```
  universal_login_experience = "classic"
}
```

» Argument Reference

The following arguments are supported:

- `universal_login_experience` - (Optional) Which login experience to use. Options include `classic` and `new`.

» auth0_tenant

With this resource, you can manage Auth0 tenants, including setting logos and support contact information, setting error pages, and configuring default tenant behaviors.

» Example Usage

```
resource "auth0_tenant" "tenant" {
  change_password {
    enabled = true
    html    = "${file("./password_reset.html")}"
  }

  guardian_mfa_page {
    enabled = true
    html    = "${file("./guardian_multifactor.html")}"
  }

  default_audience  = "<client_id>"
  default_directory = "Connection-Name"

  error_page {
    html          = "${file("./error.html")}"
    show_log_link = true
    url           = "http://mysite/errors"
  }

  friendly_name = "Tenant Name"
  picture_url   = "http://mysite/logo.png"
  support_email = "support@mysite"
  support_url   = "http://mysite/support"
  allowed_logout_urls = [
```

```
    "http://mysite/logout"
  ]
  session_lifetime = 46000
  sandbox_version  = "8"
}
```

» Argument Reference

Arguments accepted by this resource include:

- `change_password` - (Optional) List(Resource). Configuration settings for change passsword page. For details, see Change Password Page.
- `guardian_mfa_page` - (Optional) List(Resource). Configuration settings for the Guardian MFA page. For details, see Guardian MFA Page.
- `default_audience` - (Optional) String. API Audience to use by default for API Authorization flows. This setting is equivalent to appending the audience to every authorization request made to the tenant for every application.
- `default_directory` - (Optional) String. Name of the connection to be used for Password Grant exchanges. Options include `auth0-adldap`, `ad`, `auth0`, `email`, `sms`, `waad`, and `adfs`.
- `default_redirection_uri` - (Optional) String. The default absolute redirection uri, must be https and cannot contain a fragment.
- `error_page` - (Optional) List(Resource). Configuration settings for error pages. For details, see Error Page.
- `friendly_name` - (Optional) String. Friendly name for the tenant.
- `picture_url` - (Optional). String URL of logo to be shown for the tenant. Recommended size is 150px x 150px. If no URL is provided, the Auth0 logo will be used.
- `support_email` - (Optional) String. Support email address for authenticating users.
- `support_url` - (Optional) String. Support URL for authenticating users.
- `allowed_logout_urls` - (Optional) List(String). URLs that Auth0 may redirect to after logout.
- `session_lifetime` - (Optional) Integer. Number of hours during which a session will stay valid.
- `sandbox_version` - (Optional) String. Selected sandbox version for the extensibility environment, which allows you to use custom scripts to extend parts of Auth0's functionality.
- `idle_session_lifetime` - (Optional) Integer. Number of hours during which a session can be inactive before the user must log in again.
- `flags` - (Optional) List(Resource). Configuration settings for tenant flags. For details, see Flags.
- `universal_login` - (Optional) List(Resource). Configuration settings for Universal Login. For details, see Universal Login.

» Change Password Page

`change_password_page` supports the following arguments:

- `enabled` - (Required) Boolean. Indicates whether or not to use the custom change password page.
- `html` - (Required) String, HTML format with supported Liquid syntax. Customized content of the change password page.

» Guardian MFA Page

`guardian_mfa_page` supports the following arguments:

- `enabled` - (Required) Boolean. Indicates whether or not to use the custom Guardian page.
- `html` - (Required) String, HTML format with supported Liquid syntax. Customized content of the Guardian page.

» Error Page

`error_page` supports the following arguments:

- `html` - (Required) String, HTML format with supported Liquid syntax. Customized content of the error page.
- `show_log_link` - (Required) Boolean. Indicates whether or not to show the link to logs as part of the default error page.
- `url` - (Required) String. URL to redirect to when an error occurs rather than showing the default error page.

» Flags

`flags` supports the following arguments:

- `change_pwd_flow_v1` - (Optional) Boolean. Indicates whether or not to use the older v1 change password flow. Not recommended except for backward compatibility.
- `enable_client_connections` - (Optional) Boolean. Indicates whether or not all current connections should be enabled when a new client is created.
- `enable_apis_section` - (Optional) Boolean. Indicates whether or not the APIs section is enabled for the tenant.
- `enable_pipeline2` - (Optional) Boolean. Indicates whether or not advanced API Authorization scenarios are enabled.
- `enable_dynamic_client_registration` - (Optional) Boolean. Indicates whether or not the tenant allows dynamic client registration.

- `enable_custom_domain_in_emails` - (Optional) Boolean. Indicates whether or not the tenant allows custom domains in emails.
- `universal_login` - (Optional) Boolean. Indicates whether or not the tenant uses universal login.
- `enable_legacy_logs_search_v2` - (Optional) Boolean. Indicates whether or not to use the older v2 legacy logs search.
- `disable_clickjack_protection_headers` - (Optional) Boolean. Indicated whether or not classic Universal Login prompts include additional security headers to prevent clickjacking.
- `enable_public_signup_user_exists_error` - (Optional) Boolean. Indicates whether or not the public sign up process shows a user_exists error if the user already exists.

» Universal Login

`universal_login` supports the following arguments:

- `colors` - (Optional) List(Resource). Configuration settings for Universal Login colors. See Universal Login - Colors.

» Colors

`colors` supports the following arguments:

- `primary` - (Optional) String, Hexadecimal. Primary button background color.
- `page_background` - (Optional) String, Hexadecimal. Background color of login pages.

» Attribute Reference

Attributes exported by this resource include:

- `sandbox_version` - String. Selected sandbox version for the extensibility environment, which allows you to use custom scripts to extend parts of Auth0's functionality.

» auth0_user

With this resource, you can manage user identities, including resetting passwords, and creating, provisioning, blocking, and deleting users.

» Example Usage

```
resource "auth0_user" "user" {
  connection_name = "Username-Password-Authentication"
  user_id = "12345"
  username = "unique_username"
  name = "Firstname Lastname"
  given_name = "Firstname"
  family_name = "Lastname"
  nickname = "some.nickname"
  email = "test@test.com"
  email_verified = true
  password = "passpass$12$12"
  roles = [ auth0_role.admin.id ]
}

resource "auth0_role" "admin" {
    name = "admin"
    description = "Administrator"
}
```

» Argument Reference

Arguments accepted by this resource include:

- `user_id` - (Optional) String. ID of the user.
- `connection_name` - (Required) String. Name of the connection from which the user information was sourced.
- `username` - (Optional) String. Username of the user. Only valid if the connection requires a username.
- `nickname` - (Optional) String. Preferred nickname or alias of the user.
- `password` - (Optional) String, Case-sensitive. Initial password for this user. Used for non-SMS connections.
- `email` - (Optional) String. Email address of the user.
- `email_verified` - (Optional) Boolean. Indicates whether or not the email address has been verified.
- `verify_email` - (Optional) Boolean. Indicates whether or not the user will receive a verification email after creation. Overrides behavior of `email_verified` parameter.
- `phone_number` - (Optional) String. Phone number for the user; follows the E.164 recommendation. Used for SMS connections.
- `phone_verified` - (Optional) Boolean. Indicates whether or not the phone number has been verified.
- `user_metadata` - (Optional) String, JSON format. Custom fields that store info about the user that does not impact a user's core functionality.

Examples include work address, home address, and user preferences.

- `app_metadata` (Optional) String, JSON format. Custom fields that store info about the user that impact the user's core functionality, such as how an application functions or what the user can access. Examples include support plans and IDs for external accounts.
- `roles` - (Optional) Set(String). Set of IDs of roles assigned to the user.